Expert Systems with Applications 39 (2012) 11011–11021

Contents lists available at SciVerse ScienceDirect

Expert Systems with Applications

journal homepage: www.elsevier.com/locate/eswa

Multi-class particle swarm model selection for automatic image annotation ☆

Hugo Jair Escalante *, Manuel Montes, L. Enrique Sucar

National Institute of Astrophysics, Optics and Electronics, Department of Computational Sciences, Luis Enrique Erro #, 1 Tonantzintla, Puebla 72840, Mexico

ARTICLE INFO

Keywords:
Classification
Particle swarm optimization
Particle swarm model selection
Machine learning
Image annotation
Object recognition

ABSTRACT

This article describes the application of particle swarm model selection (PSMS) to the problem of automatic image annotation (AIA). PSMS can be considered a black-box tool for the selection of effective classifiers in binary classification problems. We face the AIA problem as one of multi-class classification, considering a one-vs-all (OVA) strategy. OVA makes a multi-class problem into a series of binary classification problems, each of which deals with whether a region belongs to a particular class or not. We use PSMS to select the models that compose the OVA classifier and propose a new technique for making multi-class decisions from the selected classifiers. This way, effective classifiers can be obtained in acceptable times; specific methods for preprocessing, feature selection and classification are selected for each class; and, most importantly, very good annotation performance can be obtained. We present experimental results in six data sets that give evidence of the validity of our approach; to the best of our knowledge the results reported herein are the best obtained so far in the data sets we consider. It is important to emphasize that despite the application domain we consider is AIA, nothing restricts us of applying the methods described in this article to any other multi-class classification problem. .

1. Introduction

Automatic image annotation (AIA) consists of assigning textual descriptors (labels, keywords, words) to images with the goal of supporting annotation-based image retrieval (ABIR), that is, the task of searching for images by using keywords. AIA has been recognized as one of the "*hot topics*" in the new age of multimedia information retrieval (Datta, Joshi, Li, & Wang, 2008). This is motivated by the availability of large repositories of images without any textual description associated to them. The lack of textual descriptions restricts the way the images can be searched for, as the only way to access such collections is by using content-based image retrieval (CBIR) techniques; that is, image recovery through the comparison of a sample (query) image and the stored documents. Whereas CBIR is a mature field in computer vision (Datta et al., 2008; Liu, Zhang, Lu, & Ma, 2007), CBIR methods still need of a significative amount of user interaction (e.g. for specifying query images, for drawing query-sketches, through image-category browsing and relevance feedback), which is not a desired property for any automatic image retrieval system. Therefore, effective methods are required for associating words with images (Barnard et al., 2007).

The AIA problem can be mainly approached in two different ways: at image-level and at region-level. In the former case, labels are assigned to the image as a whole, not specifying what objects correspond to which labels; in the second case, a single label is assigned to each region in segmented images, providing localization information for the objects therein. Despite both formulations provide complimentary benefits, the region-level approach provides information that is not readily available with the image-level formulation (e.g. spatial relationships between regions), which can be helpful for improving annotation performance or even for supporting image retrieval (Escalante, Montes, & Sucar, 2007; Hernandez & Sucar, 2007). For this reason we face the AIA problem at a region-level.

AIA at region-level (hereafter just AIA) consists of assigning labels, from a predefined vocabulary, to regions in segmented images. Hence, this problem can be naturally posed as a multi-class classification task, with as many classes as labels are in the vocabulary. Satisfactory results have been reported with this approach for data sets involving a few labels (Bradshaw, 2000; Szummer & Picard, 1998; Vailaya, Jain, & Zhang, 1998). However, there is little work with this technique for problems with more than 10 labels (Winn, Criminisi, & Minka, 2005). This is because the classification problem becomes more complex and accuracy decreases as the number of labels increases. Nevertheless, facing the AIA problem as a multi-class classification approach is advantageous as the best annotation results have been reported with this formulation (Bradshaw, 2000; Escalante et al., 2007; Hernandez & Sucar, 2007; Szummer & Picard, 1998; Vailaya et al., 1998; Winn et al., 2005).

☆ This project was supported by CONACyT under project grant 61335 and scholarship 205834.

* Corresponding author. Tel.: +52 2222663100x8319; fax: +52 2222663152.
E-mail address: hugojair@inaoep.mx (H.J. Escalante).

0957-4174/$ - see front matter © 2012 Elsevier Ltd. All rights reserved.
http://dx.doi.org/10.1016/j.eswa.2012.03.023

In this work, we adopt a one-vs-all (OVA) strategy for multi-class classification (Bishop, 2006). For a problem of $|C|$ classes, the OVA approach consist of building $|C|$ binary classifiers, each one is able to distinguish examples of class C_i (positive examples) from examples of any other class $C_{i \neq j}$ (negative examples). When a new instance needs to be classified, the outputs of the $C-$ classifiers are combined according to some criterion. Despite being simple, this technique has proved to be as effective as other, more complex, methods for multi-class classification, provided the best binary classifiers are used (Rifkin & Klautau, 2004). The latter finding reduces the multi-class classification problem to that of selecting the best binary classifiers for each class. There is much work from the machine learning community on this subject (known as model selection) (Bishop, 2006; Guyon e al., 2011). However, most of this work is restricted to a single model type (i. e. either selection of methods for feature selection or selection of learning algorithm, but not both) or even to a single algorithm (e. g. parameter optimization for a neural-network classifier). Furthermore, the implementation of some of these methods requires domain knowledge and significant expertise in machine learning.

This article describes the application of particle swarm model selection (PSMS) to the AIA problem under the OVA formulation. PSMS can be considered a black-box tool for the selection of effective classifiers in binary classification problems. In a nutshell, PSMS explores the space of classification models by means of particle swarm optimization and selects the model that minimizes an estimate of classification error (Escalante, Montes, & Sucar, 2009). Besides selecting classification method, PSMS selects methods for preprocessing and feature selection as well, which is a distinctive feature of the approach.

For AIA PSMS is used to select individual classification models (one for each label) for building up a multi-class classifier under OVA. This way, very effective binary classifiers will form the OVA multi-class classifier. More importantly, performing model selection for each single class allows us to consider specific methods for preprocessing, feature selection and learning for each label. This is a clear advantage over other traditional OVA classifiers in which a single learning algorithm is used for all of the classes. The selection of individual classification models by means of PSMS is particularly well suited for AIA, as the different labels require of different classification models; for example, a good classifier for the label '*sky*' is not necessarily the best classifier for the label '*building*'.

The rest of this article describes how particle swarm optimization can be used for *full model selection*, how to apply PSMS for binary classification problems and how PSMS can be used to solve multi-class classification problems, in particular, the AIA task. One should note that besides the application domain we consider is AIA, most of the methods described herein can be applied to many other binary and multi-class classification problems.

The organization of the paper is as follows. The next section introduces the background required to understand the rest of the paper. Section 3, describes the PSMS technique. Section 4 describes the OVA approach to AIA. Section 5 presents experimental results that show the validity of our approach. Section 6 outlines the conclusions derived from this work and discusses future work directions.

2. Background

This section introduces the background required to understand the rest of the paper. First we describe the AIA task, next we present the OVA approach to multi-class classification and then we describe the basic swarm optimization algorithm we have adopted for PSMS.

2.1. Automatic image annotation

AIA is a very important step towards developing more precise image retrieval systems (Datta et al., 2008; Barnard et al., 2007). However, AIA is not an easy task and, therefore, effective labeling techniques are required. The difficulty of the AIA problem is mainly due to the visual–polysemy and visual–synonymy issues. On the one hand, some regions that are visually similar may denote different concepts (e. g. '*sky*' and '*sea*'). On the other hand, regions that are visually different can be associated to the same concept (e. g. both a region with a white cow and a region with a brown one are labeled with '*cow*'). Furthermore, poor image segmentation is another complication in AIA, as automatic segmentation methods can partition a single object in more than one region. Fig. 1 illustrates the main difficulties in AIA. The difficultness of the AIA task has motivated the development of effective methods that can deal, to some extent, with specific issues; however, a complete understanding of the AIA problem is still an open topic.

The specific AIA setting we consider is as follows. Each image I_i is segmented into N_i regions, $\mathbf{r}_{1,\dots,N_i}$, visual attributes (e.g. color and area statistics) are measured from each region so that a region is represented by a vector of features. To simplify notation we denote both the jth region and the vector of features representing the jth region with $\mathbf{r}_j$. Each region $\mathbf{r}_j$ is associated with one of $|V|-$ labels (i.e. concepts, words, annotations, semantic descriptors), taken from a predefined vocabulary $V = \{v^1, \dots, v^{|V|}\}$; in particular, region $\mathbf{r}_j$ is associated with the label v_j^l that best describes its content. Thus, together labels and regions can be considered ordered pairs of the form $\left(\mathbf{r}_j, v_j^l\right) \in \mathbb{R}^d \times V$, with d the dimensionality of the feature vectors. The AIA task consists of finding a mapping from regions to labels (i.e. $f(\mathbf{r}_j) = v_j^l$), given a training set of region-label pairs, so that the obtained model can be used to predict the labels for regions, for which labels are unknown.

The predominant approach to AIA is the use of probabilistic latent variable models (Barnard et al., 2007). Instances of these sort of models are hidden Markov models (Ghoshal, Ircing, & Khudanpur, 2005), random fields (Carbonetto, de Freitas, & Barnard, 2004), correspondence latent Dirichlet allocation models (Barnard et al., 2007) and cross-media relevance models (Jeon, Lavrenko, & Manmatha, 2003). These methods are based in the formalism of probabilistic graphical models and, by introducing latent variables, they attempt to model the regions-labels joint $P\left(r_j, v_j^l\right)$ or conditional $P\left(v_j^l|\mathbf{r}_j\right)$ probability distributions (Barnard et al., 2007; Ghoshal et al., 2005; Carbonetto et al., 2004; Jeon et al., 2003; Carbonetto, 2003). The main advantage of these methods is that they require of *weakly labeled images* for training; that is, images annotated at an image-level, without any information about the explicit correspondence between regions and labels. The main problem with these methods is that their labeling accuracy is limited; also, gathering weakly annotated images is not a trivial task.

Supervised methods, on the other hand, have reported better performance than their semi-supervised[1] counterparts. However, they require of *strongly labeled images*, that is, images in which the correspondence between regions and labels is specified (Escalante et al., 2007; Hernandez & Sucar, 2007; Winn et al., 2005). Supervised methods result in higher accuracy, and, therefore, it is worthwhile spending time on creating training sets of annotated regions. Alternatively, we can take advantage of methods that use unlabeled data (Laserre, Bishop, & Minka, 2006) and web-based approaches (Fergus, Fei-Fei, Perona, & Zisserman, 2005) for building the required training data sets.

[1] In this article we use the term semi-supervised to make reference to methods trained on weakly labeled images.

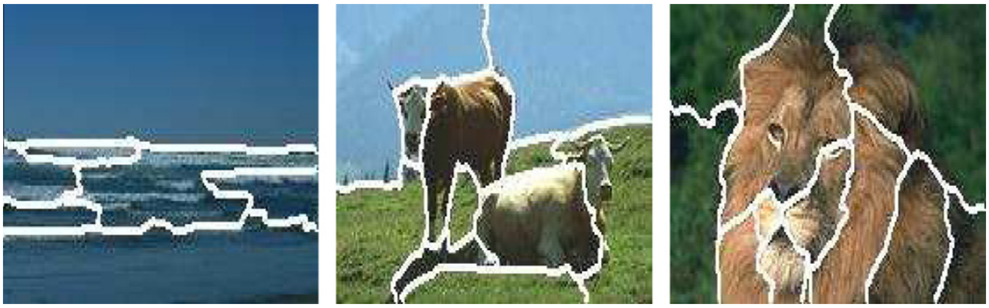

Fig. 1. The main problems in AIA: visual-polysemy (left), visual-synonymy (center) and complications due to poor segmentation (right).

2.2. One-vs-all classification

Supervised methods for AIA face the problem as one of (single-label) multi-class classification, with as many classes as labels are in the vocabulary. The goal is to find the best approximation to the map $v_j^l = f(\mathbf{r}_j)$, given a set of M – training region-labels pairs $D = \left\{\left(\mathbf{r}_1, v_1^{l_1}\right), \ldots, \left(\mathbf{r}_M, v_M^{l_M}\right)\right\}$. There are several options for facing the multi-class classification problem, a simple and widely technique is the so called OVA formulation (Bishop, 2006). OVA consists of building $|C|$ – binary classifiers, each classifier f_i is constructed by considering positive examples to regions of label C_i and negative ones to the rest $C_{j \neq i}$. When a new region $\mathbf{r}_T$ needs to be classified, each classifier f_i determines whether region $\mathbf{r}_T$ belongs to the ith class or not; then a criterion is used to select a single class for the region, starting from the outputs of the $|C|$ – classifiers.

OVA is the simplest approach one may try for multi-class classification, yet, OVA has shown comparable, and even superior, performance when compared to more complex schemes like the *all-vs-all, error-correcting output-codes* and *single-machine* approaches (Rifkin & Klautau, 2004). Also, the OVA formulation is less computationally expensive and hence this formulation is preferred over other techniques.

For AIA the OVA formulation has been applied to data sets involving a few labels (Vailaya et al., 1998; Szummer & Picard, 1998; Bradshaw, 2000). Its application to data sets with more than 10 labels is challenging because of the difficulty of the AIA task and of the inherent limitations of OVA classification. The first limitation of OVA is that some of the observations can be *ambiguously* classified, as some regions can be assigned to multiple classes simultaneously (Bishop, 2006); therefore, effective methods for selecting a single output are required. A second drawback of the OVA formulation is that the training sets for the individual classifiers are highly imbalanced (e. g. if we have 10 classes, with an equal number of examples per class, for each of the 10 – classifiers we would have 10% of positive examples and 90% of negative ones). A third limitation, related to the difficulty of the AIA task, is that of obtaining the best individual classifiers. In this respect, most researchers have considered a single classifier with fixed parameters for all of the classes. However, this is not a reliable strategy as different classes may require of different classification models. The latter is an important issue in AIA, as the classes corresponding to the labels are very different to each other; hence, different classifiers should be considered for different concepts.

The methods described in this paper can deal, to some extent, with the second and third limitations of OVA. On the one hand, we consider a fitness function that is well suited for imbalanced problems. Thus, models selected with PSMS will minimize the balanced error, instead of the usual misclassification rate. On the other hand, effective classifiers are selected for each label by means of swarm optimization. What is more, the obtained classifiers are specifically chosen for each label, which allows us modeling each concept particularly. Regarding the first limitation of OVA, we propose a simple heuristic that outperforms a widely used technique for selecting a single output for each region.

2.3. Particle swarm optimization

For this work we consider the basic PSO algorithm with adaptive inertia weight (Engelbrecht, 2006; van den Bergh, 2001); this section describes the basics of such a PSO algorithm, for a detailed description we encourage the reader to follow the references (Engelbrecht, 2006; van den Bergh, 2001; Kennedy & Eberhart, 2001). Under PSO, each solution to the problem at hand is called a particle; at each time t, each particle, i, has a position in the search space denoted by $\mathbf{x}_i^t = \langle x_{i,1}^t, x_{i,2}^t, \ldots, x_{i,d}^t \rangle$, where d is the dimensionality of the solutions; a set of particles $\mathbf{S} = \{\mathbf{x}_1^t, \ldots, \mathbf{x}_m^t\}$ is called a swarm. Particles have associated a velocity value that they use for *flying* (exploring) through the search space. The velocity of particle i at time t is given by $\mathbf{v}_i^t = \langle v_{i,1}^t, v_{i,2}^t, \ldots, v_{i,d}^t \rangle$, where $v_{i,j}^t$ is the velocity for dimension j of particle i at time t. Particles adjust their flight trajectories by using the following updating equations:

$$v_{i,j}^{t+1} = W \times v_{i,j}^t + c_1 \times r_1 \times \left(p_{i,j} - x_{i,j}^t\right) + c_2 \times r_2 \times \left(p_{g,j} - x_{i,j}^t\right) \tag{1}$$

$$x_{i,j}^{t+1} = x_{i,j}^t + v_{i,j}^{t+1} \tag{2}$$

where $p_{i,j}$ is the jth dimension of the best solution found so far by particle $\mathbf{x}_i^t$; $\mathbf{p}_i = \langle p_{i,1}, \ldots, p_{i,d} \rangle$ is often called personal best of particle $\mathbf{x}_i$.$p_{g,j}$ is the jth value of the best particle found so far in the swarm $\mathbf{S}$;$\mathbf{p}_g = \langle p_{g,1}, \ldots, p_{g,d} \rangle$ is considered the leader particle. The global and personal best particles are determined according to a fitness function that evaluates the goodness of solutions. Through $\mathbf{p}_i$ and $\mathbf{p}_g$ particles take into account individual and social information for updating their velocity and position. $c_1, c_2 \in \mathbb{R}^1$ are values that weight the contribution of the individual and social information respectively. r_1,$r_2 \sim U[0,1]$ are uniformly distributed random numbers. W is the so-called inertia weight, whose goal is to control the impact of the history of the velocities of a particle over the current velocity, influencing the local and global exploration abilities of the algorithm (Engelbrecht, 2006; van den Bergh, 2001).

The swarm is initialized randomly, taking into account restrictions on the values that each dimension can take. Then, by using Eqs. (1) and (2), particles in the swarm fly through the search space until a stop criteria is met. Usually, the process stops when either a maximum number of iterations (I) is reached or a minimum error value is obtained by a particle in the swarm; eventually, a locally-optimal solution is found.

3. Particle swarm model selection

For effectively using the OVA formulation for AIA we need to select the best classifier for each label in the annotation vocabulary.

In this context, the best classifier is the one that offers better generalization performance, that is, the one that obtains the lowest classification error in unseen data.

The task of selecting the best classifier[2] is known as model selection. Several effective model selection techniques have been proposed so far (Guyon e al., 2011; Bishop, 2006). However, most of these methods are restricted to specific classification models; also, the implementation of most of these methods requires domain knowledge or significant expertise in machine learning, which limits their applicability.

A broader view to this problem has been adopted recently, the so-called full model selection (FMS) perspective (Escalante et al., 2009). The FMS problem is as follows: given a pool of methods for preprocessing, feature selection and classification, select the combination of these that obtains the lowest classification error for a given data set. This task also includes the optimization of hyperparameters[3] for the considered methods, resulting in a vast search space to be explored, well suited for stochastic optimization techniques.

In this article we have adopted the FMS view and used particle swarm optimization (PSO) for the selection of the binary classifiers required by OVA classification for AIA; the application of PSO to FMS is known as particle swarm model selection (PSMS) (Escalante et al., 2009). We consider the FMS approach because it has the following appealing features: different model types and many methods can be considered in the selection process; it can be used by any non-expert on machine learning, as it does not require knowledge on the subject; it can be applied to any classification problem as it does not require domain knowledge; finally, and most important, very competitive models can be obtained.

The nature of FMS (i.e. a combination of combinatoric-real function optimization problem with non-smooth surface and many local minima), makes population-based search techniques well suited for this task. Because of its simplicity and proved performance PSO has been used for exploring this search space (Escalante et al., 2009; Escalante, Montes, & Sucar, 2007). PSO is preferred for FMS instead of other population-based techniques because of its simplicity and generality as no ad hoc modification was made to the based PSO algorithm for applying it to FMS. Furthermore, in previous work we have found that the way the search is guided in PSO allows PSMS to avoid overfitting, an important issue to deal with in machine learning. The performance of PSMS in binary classification has been documented elsewhere (Guyon e al., 2011; Escalante et al., 2009; Escalante et al., 2007; Guyon, Saffari, Dror, & Cawley, 2007), this paper describes its application to AIA, a multi-class classification problem.

PSO has been widely used for parameter/hyperparameter optimization. However most of the reported work is restricted to a single algorithm or even to a single parameter-hyperparameter to optimize (Kennedy & Eberhart, 2001). De Souza et al. have proposed the use of PSO for multi-class model selection (de Souza, de Carvalho, Calvo, & Ishii, 2006). They adopted the one-versus-one formulation and allowed PSO to select hyperparameters for each binary classifier. However, they only perform hyperparameter optimization for a binary support vector machine classifier (SVM). Particularly, they optimize the shrinkage and γ parameters for a fixed RBF kernel (de Souza et al., 2006). Note that in this article we consider preprocessing and feature selection methods, different learning algorithms and, even, more hyperparameters to optimize for the SVM algorithm.

3.1. PSO for classifier selection

For FMS potential solutions are models, in consequence for PSMS we represent the position of each particle i as follows:

$$\mathbf{x}_i = \langle z_{i,1}, y^1_{i,1\ldots a}, z_{i,2}, y^2_{i,2,\ldots b}, \ldots\ldots z_{i,n}, y^n_{i,1,\ldots c} \rangle \tag{3}$$

Where each $z_{i,j}$ is a binary[4] valued element that indicates the absence or presence of method j in particle (full-model) i; each entry $y^j_{i,1\ldots h}$ represents the $h-$ parameters for method j in particle i, this elements can be binary or real valued, depending on the parameters of method j. In this representation we have $n-$ methods, each with their respective hyperparameters. Under this representation any combination of preprocessing methods can be selected; however, we restrict PSMS to select a single classifier and a single feature selection method for each full model. In this respect, we have a single $z_{i,f}$ for feature selection method and another one $z_{i,c}$ for classifier. These are not binary but discrete valued, each discrete number codifies one of the available methods.

The goal of FMS is to select the full model that minimizes classification error. Accordingly, we consider a classification performance measure as fitness function for PSO. Specifically, we consider the balanced error rate (BER), defined as $BER = \frac{E_+ + E_-}{2}$, where E_+ and E_- are the misclassifications rates for the positive and negative classes, respectively. We consider BER as fitness function because this measure takes into account misclassification rates in both classes. As outlined in Section 2.2, the individual models that compose an OVA classifier approach face classification problems that are highly imbalanced. Thus, the choice of BER is well suited for the problem at hand. Furthermore, BER has been used in machine learning challenges as leading error measure for ranking participants (Guyon et al., 2007; Escalante et al., 2007).

Estimating the BER from the training data will lead to obtain models that overfit the training data (Guyon e al., 2011); thus, when such models are evaluated in a different data set, their performance would be rather poor, this issue is known as overfitting (Bishop, 2006). To avoid overfitting we estimate the BER using cross-validation (CV) instead of using the full training set. CV is a hold-out technique that provides more reliable error estimates than when the training set is used. CV consists of splitting the training data into $k-$ subsets, then $k-$ rounds of training and testing of the model are carried out. In each round, $k-1$ subsets are used for training the model and the trained model is tested in the remaining k subset. The average over the $k-$ rounds is the CV estimate of performance.

For data sets with many instances the complexity of PSMS can become intractable. Thus one must resort to heuristics that can to alleviate this issue. Consequently, each time the fitness (i.e. a model) is evaluated, we use a random subsample of the training data (still with CV), instead of using all of the data. This strategy reduces considerably processing time and at the same time helps PSMS to avoid overfitting because each time a different subset of the data is used to evaluate the model, see Escalante et al. (2009) for further details.

3.2. Classification methods considered

This section describes the the pool methods from which PSMS can choose from. We have considered the set of methods available

[2] In the rest of the paper we may refer to classifier, model and full-model indistinctively.

[3] Note that we make a distinction between parameters, those values that are estimated by the model (e.g. the weights that a neural-network learns from data), and hyperparameters, the parameters that are inherent to the model (e.g. the number of units and the learning rate used by a neural-network classifier), with the goal of highlighting the specific problems faced by PSMS.

[4] Binary elements are indeed real valued elements that take the value 1 if a threshold is exceed and 0 otherwise.

Table 1
Feature selection (FS) and preprocessing (Pre) methods available in CLOP. A brief description of the methods and their hyperparameters is presented.

Object name	Type	Hyperparameters	Description
s2n	FS	f_{max},w_{min}	Feature ranking by signal-to-noise-ratio
relief	FS	f_{max},w_{min},k_{num}	Relief feature ranking criterion
gs	FS	f_{max}	Forward feature selection with Gram-Schmidt orthogonalization
rffs	FS	f_{max},w_{min},*child*	Random forest as feature selection filter
svcrfe	FS	f_{max},*child*	Recursive feature elimination using SVM
standardize	Pre	center	Standardization of features
normalize	Pre	center	Normalization of features
$shift - scale$	Pre	offset, factor, $take_{log}$	Shifts and scales the data
$pc - extract$	Pre	f_{max}	Principal component analysis

Table 2
Available learning objects with their respective hyperparameters in the CLOP package.

Object name	Hyperparameters	Description
zarbi	None	Linear classifier
kridge	Shrinkage, coef0, degree, gamma, balance	Kernel ridge regression
naive	None	Naive Bayes
neural	Units number, shrinkage, maxiter, balance	Neural network (Netlab)
rf	Units number, mtry	Random forest
svc	Shrinkage, coef0, degree, gamma	SVM classifier

in CLOP[5] (Saffari & Guyon, 2006), a *Matlab*^{R} toolbox with implementations of several preprocessing, feature selection and classification methods; the latest version of PSMS is also available in CLOP. Table 1 shows the list of methods for preprocessing and feature selection available in the CLOP toolbox; whereas Table 2 shows the available learning algorithms. Thus, for PSMS our pool of methods to select from are those methods described in Tables 1 and 2.

We combine preprocessing, feature selection and classification methods, through the CLOP's chain object, which allows the serial combination of methods. Thus, a typical model consists of the combination of one (none) feature selection algorithm followed by a (several/none) preprocessing method, in turn followed by a learning algorithm. For example, the model given by:

$$\boldsymbol{chain}\{gs(f_{max} = 8), standardize(c = 1), neural(units = 10, s = 0.5, bal = 1, iter = 50)\}$$

uses *gs* for feature selection, selecting eight features at most, *standardization* of the data (previously centered) and a balanced *neural network* classifier with 10 units, learning rate of 0.5, and trained for 50 iterations. In consequence, the search space in FMS consists of every possible combination of methods and hyperparameters.

4. Multi-class PSMS for image annotation

For the application of PSMS to AIA we proceed as follows. We consider a training data set of M region-label pairs $D = \{(\mathbf{r}_1,w_1), \ldots, (\mathbf{r}_N,w_M)\}$, where $\mathbf{r}_j \in \mathbb{R}^d$ are vectors of features representing regions and let $w_j \in \{1, \ldots, |V|\}$ be random variables that take one of $|V| -$ values, the value of w_j is the index in the annotation vocabulary $V = \{v^1, \ldots, v^{|V|}\}$ of the label used to describe region $\mathbf{r}_j$.

Fig. 2 depicts the methodology we have adopted. The first step consists of creating $|V| -$ data subsets, $D_i = \{(\mathbf{r}_1,b_1), \ldots, (\mathbf{r}_M,b_M)\}$, with $i \in \{1, \ldots, |V|\}$ and $b_j \in [-1,1]$; in each subset D_i we assign the positive class (i.e. $b_j = 1$) to regions that belong to the ith label and the negative class (i.e. $b_j = -1$) to the rest of regions[6]. Next we apply PSMS to each of the data subsets, as described in Section 3, obtaining a specific classifier per subset. Once selected, the obtained classifiers are trained using all of the available training data.

A (test) set of regions without labels $D_T = \{(\mathbf{r}_1^T), \ldots, (\mathbf{r}_L^T)\}$ is provided and the trained classifiers are run on these data. For every test region, $\mathbf{r}_j^T$, each classifier i returns an output $f_i\left(\mathbf{r}_j^T\right) \in [-1,1]$; the positive outputs for the region are considered its *"candidate"* classes.

From these candidate classes a single label must be selected for each region. This is a complicated task as the $|V| -$ classifiers are independent to each other and, therefore, there is no guarantee that a single classifier is activated (i.e. that it returns a positive output) for each region. Additionally, it may be possible that no classifier is activated for some regions. Thus this step is crucial for obtaining positive results on AIA with OVA classification.

Diverse techniques have been proposed for assigning a single class to each test region. The widely used approach consists of selecting the class of the classifier that obtained the highest probability (Bishop, 2006). As we are not using probabilistic classifiers this method is equivalent to select the class of the classifier that obtained the the lowest classification error or the class of the classifier that obtained the highest confidence. However, since classifiers are not correlated, neither classification error nor classifiers confidence are reliable solutions (Bishop, 2006). Instead we assign a weight to each candidate class based on the distance of the test region $\mathbf{r}_j^T$ to its $k -$ nearest neighbors in the training set. The candidate class with the highest weight is assigned to the test region.

For each test instance, $\mathbf{r}_j^T$, we obtain its $k -$ nearest neighbors in the input space according to the Euclidean distance, $\eta_{\mathbf{r}_j^T} = \left\{\mathbf{n}_{j,1}^T, \mathbf{n}_{j,2}^T, \ldots, \mathbf{n}_{j,k}^T\right\}$. The set of labels associated with these $k -$ regions $w_{\mathbf{n}^T} = \{w_{\mathbf{n}^T}^1, w_{\mathbf{n}^T}^2, \ldots, w_{\mathbf{n}^T}^k\}$ is used for assigning a weight to each of the candidate classes obtained by the OVA classification. A weight is assigned to each candidate label w_l^T of region $\mathbf{r}_j^T$ as follows:

$$\rho(w_l^T) = \frac{\sum_{t=1}^{|H_l|} d^{-1}\left(\mathbf{r}_j^T, \mathbf{h}_t^l\right)}{\sum_{q=1}^{k} d^{-1}\left(\mathbf{r}_j^T, \mathbf{n}_{q,1}^T\right)} \tag{4}$$

where $H_l = \left\{\mathbf{h}_1^l, \ldots, \mathbf{h}_{|H_l|}^l\right\}$ is the subset of regions in $\eta_{\mathbf{r}_j^T}$ with label l and d^{-1} is the inverse of the Euclidean distance between x and y. As we can see the weight is assigned only to candidate labels as considered by the OVA classifier, thus acting as a filter for labels

[5] http://clopinet.com/CLOP/

[6] For some of the classes we had only 1 or 2 training regions, which imposed technical difficulties (e. g. building a classifier with a single training sample is not possible in CLOP). For this reason we generated artificial examples for classes with less than 5 examples. We obtained the mean of the available training regions for such classes and added 20 copies of this "prototype" training region to the available examples.

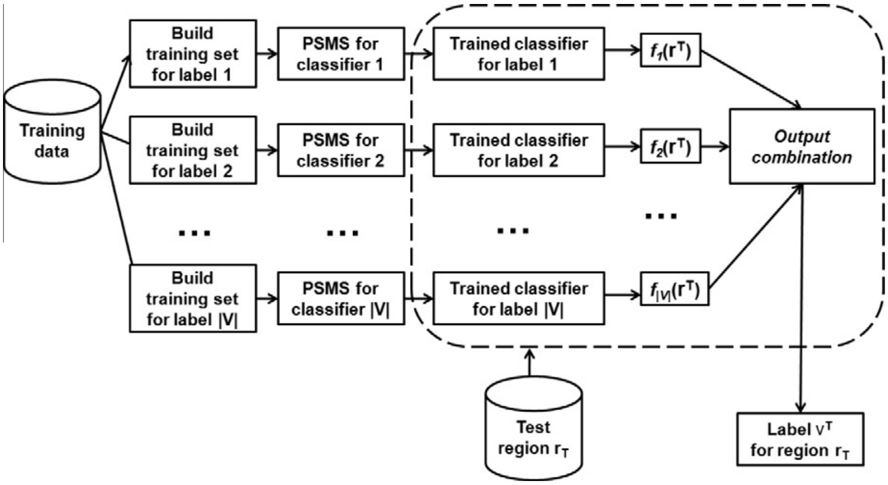
Training data
Build training set for label 1
Build training set for label 2
...
Build training set for label |V|
PSMS for classifier 1
PSMS for classifier 2
...
PSMS for classifier |V|
Trained classifier for label 1
Trained classifier for label 2
...
Trained classifier for label |V|
$f_1(r^T)$
$f_2(r^T)$
...
$f_{|V|}(r^T)$
Output combination
Test region r_T
Label v^T for region r_T

Fig. 2. Graphical diagram of the proposed approach. The training set is used for creating $|V|$ – subsets. PSMS is applied separately in each data subset, obtaining a classifier per label. For testing, the $|V|$ – classifiers classify each test instance and a heuristic is used for selecting a single label for the region.

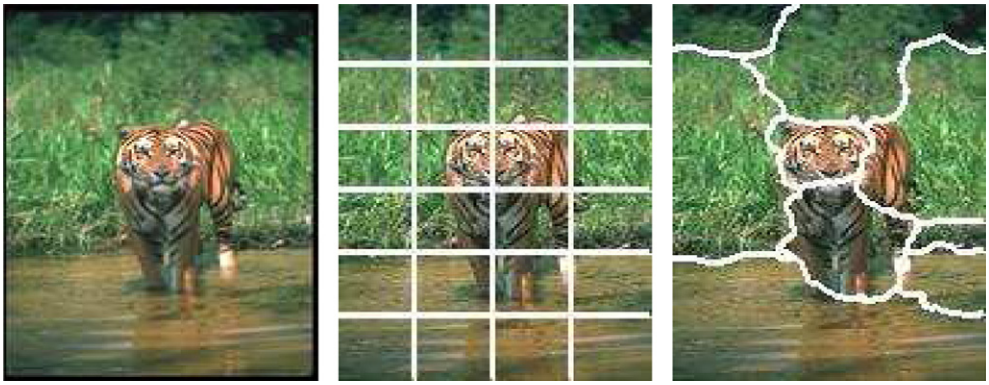

Fig. 3. Sample image from the considered Corel subsets. Left, the original image is shown; middle, the same image split in patches of fixed size; right, the same image segmented with normalized cuts (Shi & Malik, 2000).

Table 3
Statistics of the data sets considered in our experiments.

Data set	Images	Labels	Training regs.	Testing regs.
A-NCUTS	205	22	1280	728
A-GRID	205	23	3288	1632
B-NCUTS	299	38	2070	998
B-GRID	299	39	4776	2400
C-NCUTS	504	55	3328	1748
C-GRID	504	56	8064	4032

appearing in the k – nearest neighbors. The weight takes into account the proximity of the training instances as well as the repetition of these within the k – nearest neighbors. We denote this setting with PSMS-KNN, as it uses PSMS for obtaining candidate labels and a KNN-based procedure for selecting a single label for each region.

In the next section we report experimental results with PSMS-KNN. Additionally, for comparison, we present results with two baseline methods. First, we consider a widely used approach to select a single label in OVA classifiers: selecting the candidate label corresponding to the classifier that obtained the lowest (CV) classification error. Intuitively, with this strategy we will be more confident of classifiers that showed better performance during the model selection process; we call this configuration PSMS-CV. The second baseline, uses Eq. (4) for ranking all of the labels associated to the nearest neighbors of the test regions; the label with the highest weight is selected, we call this setting KNN. Note that with this strategy the candidate labels, as obtained with PSMS, are not considered.

5. Experiments and results

For the experiments reported in this section we consider 6 subsets[7] taken from the Corel^{TM} collection, a widely used benchmark for the evaluation of AIA methods (Datta et al., 2008; Barnard et al., 2007; Carbonetto et al., 2004; Jeon et al., 2003; Carbonetto, 2003). Images in these data sets have been segmented into patches of equal size (GRID) or by using the normalized cuts (NCUTS) algorithm (Shi & Malik, 2000); all of the regions are manually annotated according to different vocabularies. Each data set has predefined partitions of training and testing sets that have been used elsewhere (Carbonetto et al., 2004; Hernandez & Sucar, 2007; Escalante et al., 2007). Fig. 3 shows sample images from data sets we consider and Table 3 shows statistics of these data sets. The following features are used to represent each region: region area, mean and standard deviation in the x

[7] http://www.cs.ubc.ca/~pcarbo/

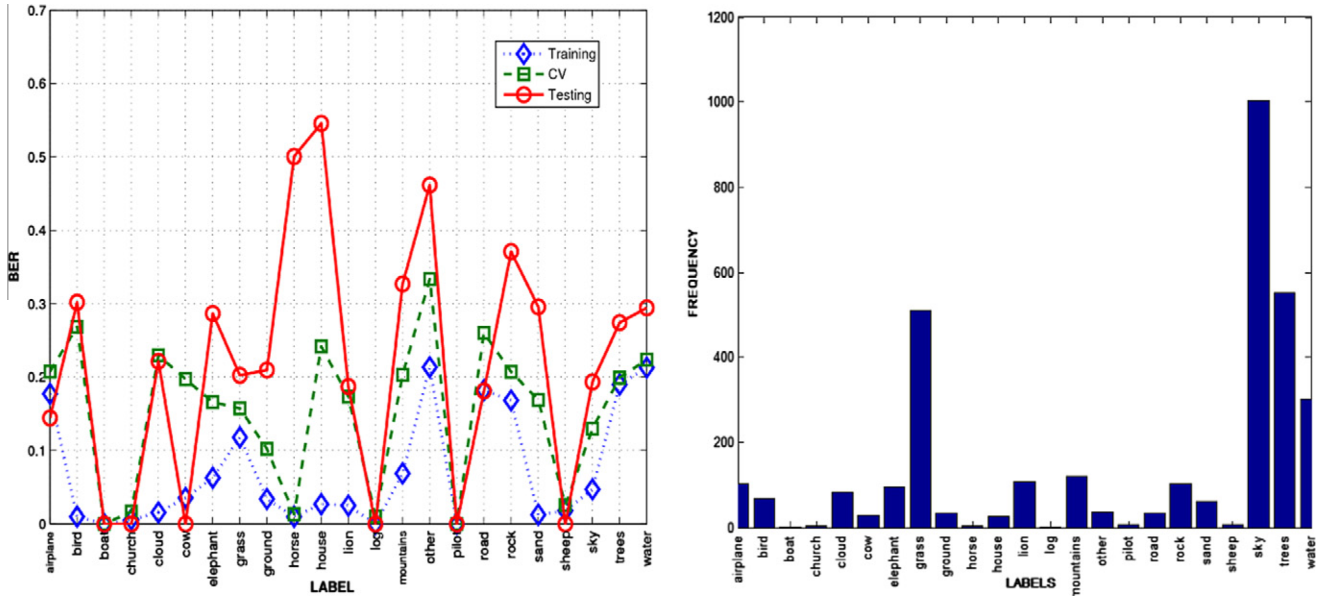

Fig. 4. Left: training, CV and test BER of the models selected with PSMS for the A-GRID data set. The best annotation results were obtained in this data set, see Section 5.3. Right: number of training examples available in this set for each class.

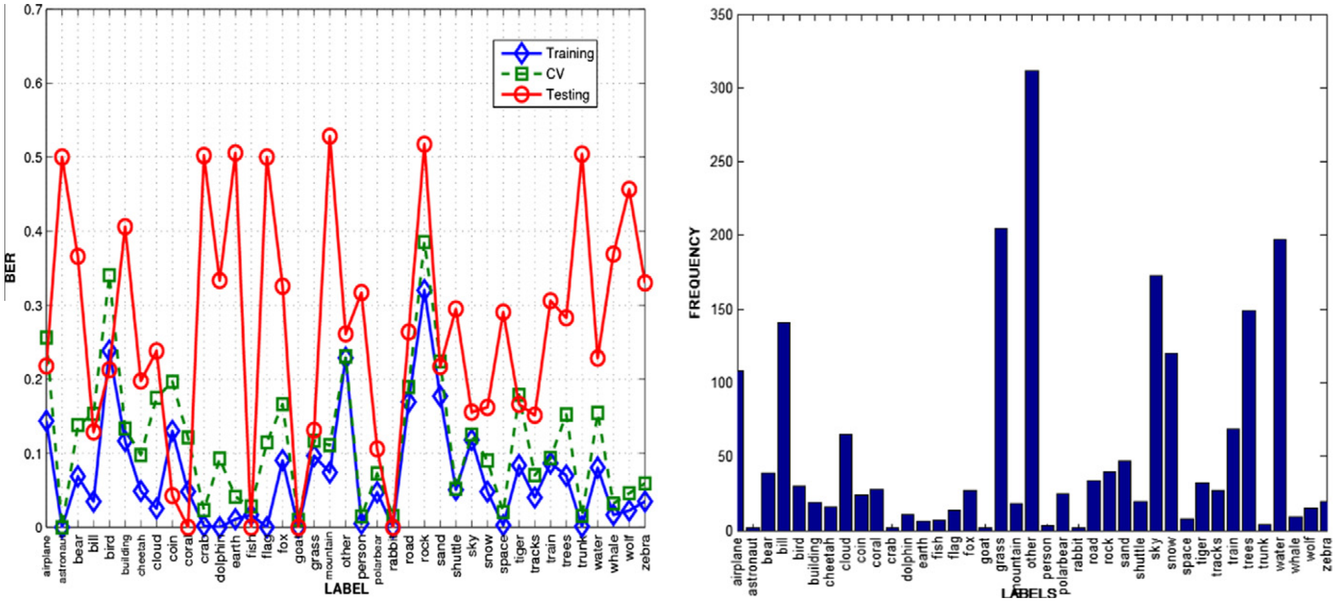

Fig. 5. Left: training, CV and test BER of the models selected with PSMS for the B-NCUTS data set. Right: number of training examples available in this set for each class.

and *y* axis, boundary/area, convexity, average, standard deviation and skewness in the CIE-Lab color space, for a total of 16 features.

In each data set we proceed as follows. For each label we run PSMS[8] for 50 iterations, using the training partition, to select a classifier for the label; in average it took about 3 min applying PSMS for selecting each classifier. Once selected, the model is trained using all of the training data. The trained classifiers are used to classify the regions in the test partition. Then, a single label is selected for each test region, from its set of candidate labels, according to the three techniques described above.

We have divided the results in three parts: first, in Section 5.1, we analyze the performance of the individual classifiers selected with PSMS; next, in Section 5.2, we analyze the labeling accuracy in the set of candidate labels obtained with PSMS; finally, in Section 5.3, we analyze the multi-class performance of the OVA classifier composed of models selected with PSMS.

5.1. Individual performance of classifiers

We start by analyzing the individual models selected with PSMS; our goal is to evaluate how good the classifiers selected by PSMS are. For space limitations we consider for this analysis the A-GRID and B-NCUTS data sets only as these are the data sets in which we obtained the best and the worse results respectively,

[8] Code with the latest PSMS implementation and related documentation can be found at: http://ccc.inaoep.mx/~hugojair

Table 4
Models selected with PSMS for each of the classes in the A-NCUTS data set.

Class	Preprocessing	Feature selection	Learning
‘*airplane*’	Standardize, normalize, shift-scale	pc-extract	SVC-RBF
‘*bird*’	Standardize, normalize, shift-scale	s2n	SVC-RBF
‘*boat*’	Standardize, normalize, shift-scale	gs	SVC-RBF
‘*church*’	Standardize, normalize, shift-scale	relief	SVC-RBF
‘*cow*’	Standardize, normalize, shift-scale	s2n	SVC-RBF
‘*elephant*’	Shift-scale	pc-extract	SVC-RBF
‘*grass*’	Standardize, shift-scale	s2n	SVC-RBF
‘*ground*’	Standardize, normalize, shift-scale	s2n	Zarbi
‘*horse*	Normalize	s2n	SVC-RBF
‘*house*’	Standardize, normalize, shift-scale	gs	SVC-RBF
‘*lion*’	Normalize	gs	SVC-RBF
‘*log*’	Standardize, normalize, shift-scale	gs	SVC-RBF
‘*mountains*’	Standardize, normalize, shift-scale	relief	SVC-RBF
‘*other*’	Standardize, normalize	s2n	SVC-RBF
‘*pilot*’	–	Relief	SVC-RBF
‘*road*’	Standardize, normalize, shift-scale	s2n	SVC-RBF
‘*rock*’	Standardize, normalize	s2n	SVC-RBF
‘*sand*’	Standardize, normalize, shift-scale	gs	SVC-RBF
‘*sheep*’	Standardize, normalize, shift-scale	pc-extract	SVC-RBF
‘*sky*’	Standardize, shift-scale	s2n	SVC-RBF
‘*trees*’	Standardize, normalize, shift-scale	gs	Zarbi
‘*water*’	Standardize, normalize, shift-scale	gs	SVC-RBF

see Section 5.3. For each label, we use PSMS for selecting a classifier, the selected model is trained using the full training set. Using the trained model we classify (i) the training regions, (ii) the training regions according CV, and (iii) the test regions, calculating the BER accordingly. Fig. 4 shows the BER obtained in the training, CV and test regions by each of the classifiers for the A-GRID data set as well as the number of training regions per label.

From Fig. 4 (left) we can see that most classifiers obtained a BER of 30% or less, which is a very positive result given the highly imbalanced data sets. Overfitting seems not to be a problem for most of the classes as the CV error is close to the test set error. Although, for the labels ‘*horse*’ and ‘*house*’ it is a serious problem. This is mainly due to number of examples available for these classes. Note that (Fig. 4, right) we have more than 1000 examples of the label ‘*sky*’ and six classes with less than 10 examples. For the classes ‘*horse*’ and ‘*house*’ we have 4 and 26 positive examples, respectively. This means that we have 99.8% and 99.2% negative examples and only 0.12% and 0.8% of positive ones, for these labels, respectively. Therefore, even when we obtained a high test error this is not surprising. On average the performance of models selected with PSMS is very competitive.

Fig. 5 shows the BER of the models selected for the B-NCUTS. From this plot we can see that the performance of 8 out of 38 classifiers is close to 50% (the worst possible result). This is, again, due to the imbalanced data set. However, it is also due to the large number of classes in this data set (i.e. 38). For this data set, overfitting is a serious problem for the following classes: ‘*astronaut*’, ‘*crab*’, ‘*earth*’, ‘*fox*’, ‘*mountain*’, ‘*rock*’, ‘*trunk*’ and ‘*wolf*’. The number of training examples for this classes is of 2, 2, 6, 27, 18, 40 and 4, respectively. Thus, the percentages of positive examples for each of these data sets is of 1%, 1%, 1.25%, 1.3%, 0.8%, 1.9% and 1.1%, respectively. Therefore, the overfitting problems with PSMS are directly related to the number of instances available per class.

With the goal of giving insight into the type of models selected by PSMS, Table 4 shows the models selected with PSMS for each label in the A-NCUTS data set. For clarity, we do not show the hyperparameters selected for each model, although we emphasize that these were different for each of the classes. Also for clarity, we show the models selected for the A-NCUTS data set, as this is the data set with less labels. Fig. 6 shows the BER obtained by each of the selected models.

As we can see, there is a strong preference for SVM classifiers with a RBF kernel. Also, for most of the classes the triplet of *’standardize,normalize,shift-scale’* was selected for preprocessing. The feature selection method was the only model type that varied through the classes. It is interesting that for the ‘*ground*’ and ‘*trees*’ classes PSMS selected a simple linear classifier; the selection of these classifiers gives evidence that the respective classes are linearly separable. It is also interesting that these linear classifiers outperformed many models, even other classifiers for labels with more training examples. For example, there are 30 training examples for

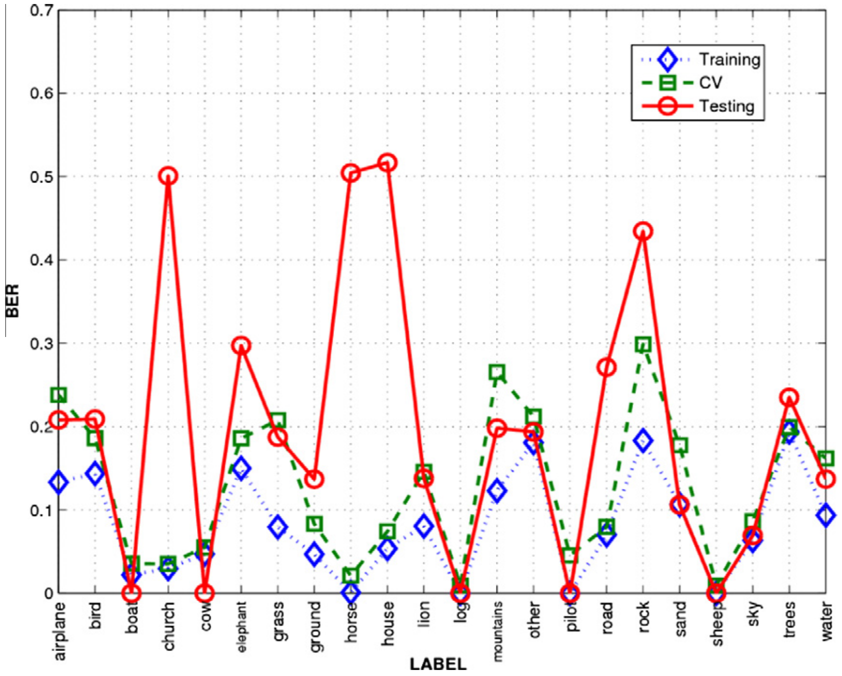

Fig. 6. Training, CV and test BER of the models selected with PSMS for the A-NCUTS data set.

Table 5
Comparison of accuracy in the candidate labels for PSMS-CV, KNN and PSMS-KNN. We show accuracy and between parentheses the first number shows the average number of candidate labels for each test region and the second value shows the maximum number of candidate labels for a region in each data set.

Data set	PSMS-CV	KNN	PSMS-KNN
A-NCUTS	78.02% (2.96–7)	89.42% (6.44–13)	72.39% (2.32–7)
A-GRID	69.06% (3.02–8)	90.50% (4.95–12)	68.75% (2.22–8)
B-NCUTS	62.83% (4.14–10)	80.96% (7.49–14)	59.72% (2.68–10)
B-GRID	66.25% (4.82–13)	77.54% (5.94–15)	61.42% (2.70–13)
C-NCUTS	74.31% (4.94–13)	83.87% (7.53–16)	68.31% (2.95–13)
C-GRID	73.54% (6.34–17)	83.63% (5.99–17)	71.53% (3.96–17)
Average	70.67% (4.37–11.33)	84.32% (6.39–14.50)	67.02% (2.80–11.33)

the label *'ground'* and 114 for the label '*water*' for which a SVM was used; however, the performance of both classifiers is very similar.

5.2. On the quality of candidate labels

In this section we evaluate the candidate labels for test regions as obtained with PSMS. The goal is to evaluate how effective classifiers are for including the correct label for test regions in their set of candidate labels; based on this analysis we can set an upper bound in the maximum accuracy we can get in the multi-class task. For this experiment, we consider that a region is correctly classified if the correct label appears in the set of candidate labels for that region; we estimate accuracy as the percentage of regions that were correctly classified. We include results with the KNN method for comparison.

For PSMS-CV the candidate labels for a region are the classes corresponding to the classifier that provided a positive output for the region, see Section 4. For KNN we consider as candidate labels to the labels assigned to the top[9] $k = 20$ nearest neighbors of the test regions. For PSMS-KNN we consider the set of labels that appear in both candidate sets, that of PSMS-CV and that of KNN. Table 5 shows results of this comparison.

As we can see the highest accuracy is obtained by the KNN method; however, we emphasize that we are considering the labels of the top -20 nearest neighbors to the test regions. In average, 6.39 out of 20, are different (i.e. some labels are repeated), which represents the 18% of the average number of labels in the annotation vocabularies. The average maximum number of candidate labels for a region is large, almost 15 labels, which represents the 39.1% of the vocabulary. Having a large number of candidate labels makes difficult the selection of a single correct label, and hence affects the annotation performance. For high values of k, say $k = 80$, we have almost perfect accuracy considering candidate labels with KNN, however, almost all of the labels in the vocabulary are considered candidate labels.

For the classifiers selected with PSMS, accuracy in the set of candidate labels is not that good. However, we can see that the number of candidate labels is rather small. For PSMS-CV we have in average 4.37 candidate labels which represents 12% of the average size of the vocabulary. While for PSMS-KNN we have in average 2.8 candidate labels only, this represents the 7.3% of the total of available labels. This is a clear advantage of methods that involve PSMS as for selecting a single label for each region we will have to choose from a small set of candidate labels. Therefore, the probability of selecting the correct label is large. For example, for PSMS-KNN by making random-uniform selections we will have a probability of 0.36 of picking the correct label, while for KNN this probability is reduced to 0.14, since we must choose from a set of a higher cardinality.

From Table 5 we can also appreciate that by filtering the labels returned by PSMS-CV using Eq. (4), i.e. PSMS-KNN, we have a small loss of accuracy (3% in average). This result shows that the overlap between candidate classes of PSMS-CV and KNN is high and that PSMS-CV returns better labels, if accuracy is amortized by the number of candidate labels. Results from this experiment reflect the quality of the models selected with PSMS. Accuracy is not as good as that of KNN we are considering only 2–4 candidate labels; opposed to KNN in which 20 labels are considered. Accuracy of KNN decreases as we consider less labels and the number of candidate labels increase as we consider more labels. Therefore, there is a dependence on the value of k. For PSMS we always obtain good classifiers and a few candidate classes only.

5.3. Multi-class classification performance

In the next experiment we evaluate the annotation accuracy of the techniques we adopted for selecting a single label for each region. In this section we say a region is correctly annotated if the single label selected by a method, from the set of candidate labels, is the correct one. Fig. 7 shows results of this experiment.

From this figure, we can see that the best results are obtained by the labels selected with PSMS-KNN. KNN outperforms PSMS-KNN in a single data set (A-GRID); however, the difference is less than 1%, and then it can be neglected. In average, PSMS-KNN outperforms *KNN* by 2.7%, considering 41% less of candidate labels (i. e. 6.9 to 2.8). This is a very interesting result as the upper bound in accuracy for KNN is 84% compared to that of PSMS-KNN which is of 67%, this is a difference of 17.3%. This result, again, shows that classifiers selected with PSMS are better than our strong baseline (KNN). Furthermore, results show that our technique for the selection of a single label outperforms the widely used approach (i.e. PSMS-CV). Note that the worst results with PSMS-KNN are obtained in the B-NCUTS data set. As mentioned in Section 5.1 the reduction in performance is due to the high classes-imbalance and to the number of classes considered. These issues also affect the performance of KNN.

In order to compare our results with state of the art methods we considered the methods proposed by Carbonetto et al. (2004) and Carbonetto (2003)) and compared them with ours. These are semi-supervised methods trained using weakly-labeled images. These methods were evaluated in the same data sets we have considered; to the best of our knowledge, these are the only methods for which region-level accuracy has been evaluated.

We used the parameter configurations proposed by the authors, and even used the same code and error measure, see the caption of Fig. 8. For this experiment, we consider the subsets A-NCUTS and A-GRID due to space limitations and because these data sets have been considered in other works (Carbonetto et al., 2004; Carbonetto, 2003; Hernandez & Sucar, 2007; Escalante et al., 2007). Fig. 8 shows results of the experiment.

In Fig. 8 error is calculated as follows:

$$e = \frac{1}{N}\sum_{n=1}^{N}\frac{1}{M_n}\left(1 - \delta\left(\bar{a}_{nu} = a_{nu}^{max}\right)\right) \qquad (5)$$

where M_n is the number of regions in image n, N is the number of images in the collection; and δ is a function that is 1 if the predicted annotation a_{nu}^{max} is the same as the true label $\bar{a}_{nu}$. The semi-supervised methods were run for 10 trials, while ours where run a single trial, since for all of our experiments the results with our methods do not vary each trial.

PSMS-CV outperforms all of the semi-supervised techniques, although it performs worst than KNN; showing that KNN is a

[9] We considered this value because accuracy increases as more labels are considered. A value of $k = 20$ provides a strong baseline to compare our approach. Furthermore, the value $k = 20$ gives better results when combined with PSMS (i. e. PSMS + KNN) than other values we have tried. Accuracy for values higher than $k = 20$ does not increase significantly, although the number of candidate labels does.

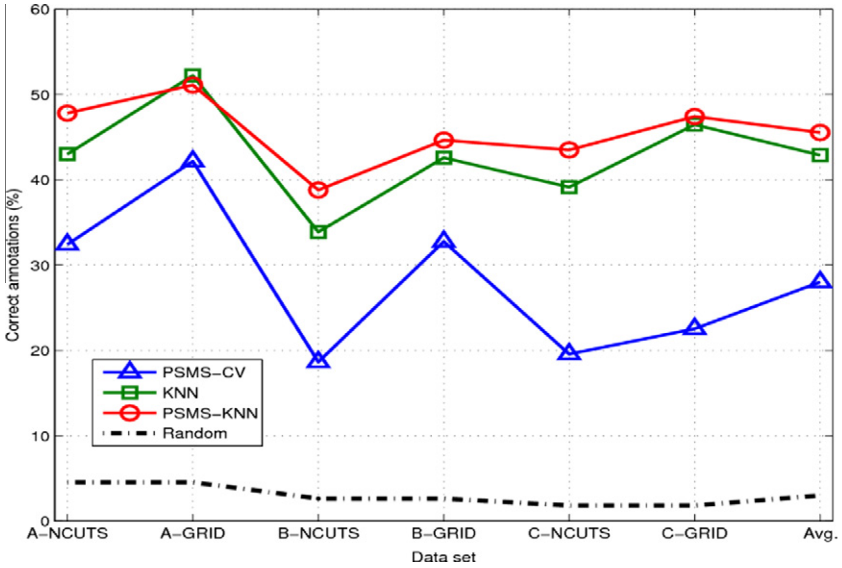

Fig. 7. Annotation accuracy of the considered methods: PSMS-CV (line, rhombic marker), KNN (dashed line, square marker) and PSMS-KNN (dotted line, circle marker); the lower dotted-dashed line represents the accuracy we would obtain if we would pick labels from the predefined vocabulary at random.

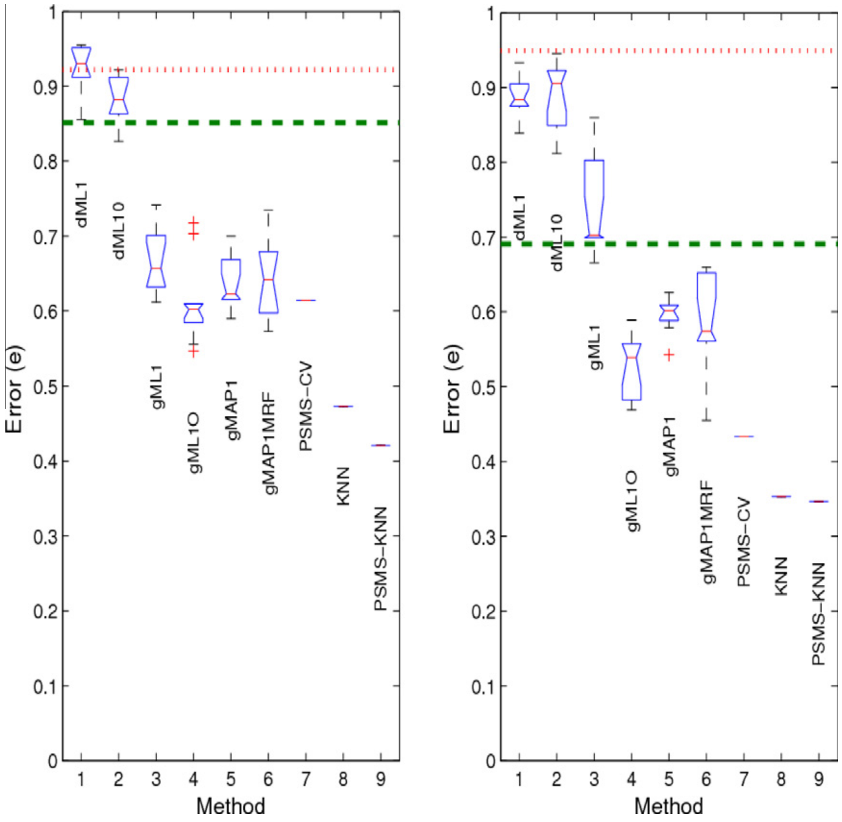

Fig. 8. Annotation error (see Eq. (5)) of PSMS-CV, KNN and PSMS-KNN compared to other state of the art methods: dML1 (Barnard et al., 2007); dML1O, gML1, gMLO, gMAP1 (Carbonetto, 2003); gMAP1MRF (Carbonetto et al., 2004). A box-and-whisker plot is used. The central box represents the values from the 25 to 75 percentile, the line in the middle of the box shows the average error; outliers are shown as separate points. Results are shown for the A-NCUTS data set (left) and for the A-GRID data set (right). The dotted line represents the error obtained if labels were chosen randomly, while the dashed line represents a method that always assigns the same label to all regions.

strong baseline. PSMS-KNN outperforms all other approaches in both subsets. The difference with the semi-supervised methods is large and illustrates the advantages of using strongly-labeled images for training.

Results reported in Sections 5.2 and 5.3 are superior to other supervised techniques that use spatial relationships information (Hernandez & Sucar, 2007) and word co-occurrence (Escalante et al., 2007) for improving AIA performance. To the best of our knowledge the results presented in this paper are the best reported so far for the considered data sets. Our results cannot be compared directly with other methods proposed in the AIA literature, since with exception (Carbonetto et al., 2004; Carbonetto, 2003;

Hernandez & Sucar, 2007; Escalante et al., 2007) no method has been properly evaluated in terms of region-level AIA performance. Most techniques have been evaluated by looking at their image level performance only (Barnard et al., 2007; Ghoshal et al., 2005; Jeon et al., 2003).

6. Conclusions

We have described the use of particle swarm model selection (PSMS) for the task of region-level automatic image annotation (AIA). PSMS is the application of the basic particle swarm optimization (PSO) algorithm for the problem of full model selection in binary classification. For applying PSMS to AIA (a multi-class classification task) we have adopted the one-vs-all (OVA) strategy, which consists of combining the outputs of several binary classifiers, each one associated to a single label/class. Accordingly, PSMS is used to select the individual models that compose the OVA classifier, obtaining specific models for each label. Also, we have proposed a new technique for combining the outputs of the individual classifiers under OVA.

Experimental results in six data sets give evidence of the validity of the proposed techniques. In particular, we confirmed that classifiers selected with PSMS were very effective for distinguishing regions of their corresponding labels. When testing the individual models, we found that, in average, few classifiers were activated yet achieving satisfactory performance. When evaluating AIA, we found that the proposed technique for output combination in OVA, obtained very good performance, outperforming KNN and a traditional technique. Also, the proposed method outperforms by a large margin a variety of semi-supervised AIA techniques. To the best of our knowledge, the results reported herein are superior to that described elsewhere, using supervised or semi-supervised methods, where the same data sets have been considered.

Despite the application domain of PSMS is AIA, nothing restricts us applying the methods described in this article to any other multi-class classification problem. The main benefits of adopting a similar strategy are evident: one can obtain effective individual classifiers and very good multi-class performance, in an acceptable time; specific methods for preprocessing, feature selection and classification are considered for each class, thus modeling classes particularly; and no specialized knowledge is required for using our methods.

Several future work directions can be outlined to extend the proposed methods. From the point of view of swarm intelligence, we can adopt more sophisticated or specialized PSO implementations that can improve the search process, for example, the use of discrete or hybrid PSO algorithms is a direct extension for PSMS; by adopting multi-swarm PSO strategies we can improve the search process while making it more efficient; a parallelization of the algorithm is also feasible. From a machine learning perspective the method can be improved as well, for example, by introducing prior domain knowledge for effectively dividing the search space in a multi-swarm PSO implementation; by incorporating a penalty term into the fitness function so that the aptitude of the individual classifiers also depend on their multi-class performance; by considering the correlations between the individual classifiers for selecting the final output of OVA classification; and by implementing PSMS in other machine learning toolboxes (e.g. the popular WEKA toolbox).

References

Barnard, K., Duygulu, P., de Freitas, N., Forsyth, D., Blei, D., & Jordan, M. I. (2007). Matching words with pictures. *Journal of Machine Learning Research., 3*(Mar), 1107–1135.

Bishop, C. (2006). *Pattern recognition and machine learning*. Springer.

Bradshaw, B. (2000). Semantic based image retrieval: a probabilistic approach. In: *Proc. of the 8th ACM international conference on multimedia* (pp. 167–176). California, USA

Carbonetto, P. (2003). Unsupervised statistical models for general object recognition, M.S. thesis, University of British Columbia.

Carbonetto, P., de Freitas, N., & Barnard, K. (2004). A statistical model for general context object recognition. In: *Proc. of the 8th european conference on computer vision* (LNCS Vol. 3021, pp. 350–362). Springer, Prague, Czech Republic.

Datta, R., Joshi, D., Li, J., & Wang, J. Z. (2008). Image retrieval: ideas, influences, and trends of the new age. *ACM Computing Surveys, 40*(2), 1–60.

de Souza, B.F., de Carvalho, A., Calvo, R.,& Ishii, P. (2006). Multiclass SVM model selection using particle swarm model selection. In: *Proc. of the 6th international conference on hybrid intelligent systems* (pp. 31–35). Rio De Janeiro, Brazil.

Engelbrecht, A. P. (2006). *Fundamentals of computational swarm intelligence*. Wiley.

Escalante, H.J., Montes, M., & Sucar, E. (2007). PSMS for neural networks in the IJCNN 2007 ALvsPK. In: *Proc. of 20th international joint conference on neural networks* (pp. 1191–1197). Orlando, FL, USA.

Escalante, H. J., Montes, M., & Sucar, L. E. (2007). Word co-occurrence and Markov random fields for improving automatic image annotation. In: *Proc. of the 18th British machine vision conference* (Vol. 2, pp. 600–609), Warwick, UK.

Escalante, H. J., Montes, M., & Sucar, L. E. (2009). Particle swarm model selection. *Journal of Machine Learning Research, 10*(Feb), 405–440.

Fergus, R., Fei-Fei, L., Perona, P., & Zisserman, A. (2005). Learning object categories from Googles image search. In: *Proc. of the 10th international conference on computer vision* (pp. 1816–1823). Beijing, China.

Ghoshal, A., Ircing, P., & Khudanpur, S. (2005). Hidden Markov Models for automatic annotation and content-based retrieval of images and video. In: *Proc. of the 28th international ACM-SIGIR conference on research and development in information retrieval* (pp. 544–551). Salvador, Brazil.

Guyon, I., Cawley, G., Dror, G., & Saffari, A. (2011). *Hands-on pattern recognition, challenges in machine learning series,* Vol. 1. Microtome Publishing, Brookline, Massachusetts.

Guyon, I., Saffari, A., Dror, G., & Cawley, G. (2007). Agnostic learning vs prior knowledge challenge. In: *Proc. of 20th international joint conference on neural networks* (pp. 1232–1238). Orlando, FL, USA.

Hernandez, C., & Sucar, L. E. (2007). Markov random fields and spatial information to improve automatic image annotation. *Proc. of the 2007 Pacific-Rim Symposium on Image and Video Technology, LNCS* (Vol. 4872, pp. 879–892). Santiago,Chile: Springer.

Jeon, J., Lavrenko, V., & Manmatha, R. (2003). Automatic image annotation and retrieval using cross-media relevance models. In: *Proc. of the 26th international ACM-SIGIR conference on research and development on information retrieval* (pp. 119–126). Toronto, Canada.

Kennedy, J., & Eberhart, R. (2001). *Swarm intelligence*. Morgan Kaufmann.

Laserre, J., Bishop, C., & Minka, T. (2006). Principled hybrids on discriminative and generative models. In: *Proc. of the conference on computer vision and pattern recognition* (pp. 87–94). New York, USA.

Liu, Y., Zhang, D., Lu, G., & Ma, W. (2007). A survey of content-based image retrieval with high-level semantics. *Pattern Recognition, 40*(1), 262–282.

Rifkin, R., & Klautau, A. (2004). In defense of one-vs-all classification. *Journal of Machine Learning Research, 5*(Jan), 101–141.

Saffari, A., & Guyon, I. (2006). Quickstart Guide for CLOP, Tech. rep., Graz University of Technology and Clopinet, Graz, Austria.

Shi, J., & Malik, J. (2000). Normalized Cuts and Image Segmentation. *IEEE Trans. on PAMI, 22*(8), 888–905.

Szummer, M., & Picard, R. (1998). Indoor–outdoor image classification. In: it Proc. of the workshop on content-based access to image and video databases (pp. 42). Washington, DC, USA.

Vailaya, A., Jain, A., & Zhang, H. (1998). On image classification: city versus landscape. *Pattern Recognition, 31*, 1921–1936.

van den Bergh, F. 2001. *An analysis of particle swarm optimizers.* PhD thesis, University of Pretoria, Sudafrica.

Winn, J., Criminisi, A., & Minka, T. (2005). Object categorization by learned universal visual dictionary. In: *Proc. of the 10th international conference on computer vision* (pp. 1800–1807). Beijing, China.