IDIAP RESEARCH REPORT

LEARNING FROM IMAGES WITH CAPTIONS USING THE MAXIMUM MARGIN SET ALGORITHM

Jie Luo Francesco Orabona Barbara Caputo
Vittorio Ferrari

Idiap-RR-30-2011

AUGUST 2011

Centre du Parc, Rue Marconi 19, P.O. Box 592, CH - 1920 Martigny
T +41 27 721 77 11 F +41 27 721 77 12 info@idiap.ch www.idiap.ch

Learning from Images with Captions Using the Maximum Margin Set Algorithm

Luo Jie, Francesco Orabona, Barbara Caputo and Vittorio Ferrari

Abstract—A large amount of images with accompanying text captions are available on the Internet. These are valuable for training visual classifiers without any explicit manual intervention. In this paper, we present a general framework to address this problem. Under this new framework, each training image is represented as a bag of regions, associated with a set of candidate labeling vectors. Each labeling vector encodes the possible labels for the regions of the image. The set of all possible labeling vectors can be generated automatically from the caption using natural language processing techniques. The use of labeling vectors provides a principled way to include diverse information from the captions, such as multiple types of words corresponding to different attributes of the same image region, labeling constraints derived from grammatical connections between words, uniqueness constraints, and spatial position indicators. Moreover, it can also be used to incorporate high-level domain knowledge useful for improving learning performance. We show that learning is possible under this weakly supervised setup. Exploiting this property of the problem, we propose a large margin discriminative formulation, and an efficient algorithm to solve the proposed learning problem. Experiments conducted on artificial datasets and two real-world images and captions datasets support our claims.

Index Terms—Weakly supervised learning, candidate labeling sets, images and captions, multi-class and multi-label classification, convex and non-convex optimization, large margin classifiers

1 Introduction

A huge amount of images with accompanying captions are available on the Internet. Websites selling various items such as houses and clothing provide photographs of their products along with concise descriptions. Online newspapers (e.g. news.yahoo.com) have pictures illustrating events and comment them in the caption. These news websites are very popular because people are interested in other people, especially if they are famous (fig. 1). This motivates the recent interest in using captioned images for training visual classifiers. Exploiting the latent associations between images and text can lead to a virtually infinite source of training annotations, without any explicit manual intervention. The learned model can then be used in a variety of Computer Vision applications, including face recognition, image search engines, and to annotate new images for which no caption is available.

There have been several works that study this problem on different applications and from different perspectives. Previous works have focused on associating names [4], [26] and verbs [29] in the captions to the faces and body poses of people in news images, on learning character naming systems from TV series using scripts [16] and screenplays [10], on learning scene classification models from tagged photos [3], [24], [41], and on learning object recognition models from an online nature encyclopedia [40]. All these can be considered as weakly supervised learning problems, because each segment, face, pose, or object in the image is only indirectly, ambiguously labeled by the words in the captions.

The above tasks are more challenging than standard supervised learning tasks due to the *correspondence ambiguity* problem: it is not known beforehand which part of the image corresponds to which part of the caption. Moreover, not everything mentioned in a natural text caption appears in the image, and, vice-versa, not everything in the image is mentioned by the caption. This is different from using tags which are guaranteed to describe the image, as in the Corel database [3]. On the other hand, natural language descriptions contain rich semantic information about the relations between different image regions and labels. For example, in fig. 1 (left), knowing what "waves" (verb) means would reveal who of the two imaged persons is "Barak Obama" (subject). The other way around, knowing who is "Barak Obama" would deliver a visual example for the "waving" pose [29]. This connection between the name and the verb can be exploited to constrain the labeling: if a region is labeled by the name Barak, then it must also be labeled by "waving". A labeling like "Barack-standing" is not valid given the caption. The caption sometimes enables to impose also other constraints. For instance, we know that Federer cannot appear twice in

- *L. Jie is with the Idiap Research Institute, CH-1920 Martigny, Switzerland and the Swiss Federal Institute of Technology in Lausanne (EPFL), CH-1015 Lausanne, Switzerland. E-mail: jluo@idiap.ch*
- *F. Orabona is with the DSI, Università degli Studi di Milano, 20135 Milano, Italy. E-mail: francesco@orabona.com*
- *B. Caputo is with the Idiap Research Institute, CH-1920 Martigny, Switzerland. E-mail: bcaputo@idiap.ch*
- *V. Ferrari is with the CALVIN research group, Computer Vision laboratory, Swiss Federal Institute of Technology in Zurich (ETHZ), CH-8092 Zurich, Switzerland. E-mail: ferrari@vision.ee.ethz.ch*

an image, so no two image regions can take the same label "Federer". As another example, captions sometimes contain spatial position indicators. In the example of fig. 2, "Chervynsky" cannot be a valid label for the person in the middle. Such constraints can be used to prune the space of possible labelings, which facilitates learning [4], [26]. To the best of our knowledge, all existing algorithms are designed to explicitly incorporate a particular type of constraint. This means the algorithm has to be redesigned in order to integrate a new type of constraint.

In this paper, we propose a general, weakly supervised learning framework to model the problem of learning from images with captions. In this framework, each training image is represented as a bag of regions, and is associated with a set of *candidate labeling vectors*. Each candidate labeling vector encodes a possible labeling of all regions, with only one candidate labeling being fully correct. The set of candidate labeling vectors can be generated automatically from the captions using natural language processing (NLP) tools. This framework provides a unified way to include many types of constraints.

The contributions of this paper are: (i) we present a general framework which provides a principled way to include various types of constraints generated from the captions; (ii) we provide a theoretical analysis that justifies our framework, showing that under certain conditions it is possible to train classifiers even under this weakly supervised setup. Exploiting this property of the setup, we also propose a large margin discriminative formulation with an efficient stochastic gradient descent algorithm to optimize it; (iii) we present experiments on artificial datasets and two real-world datasets of images and captions. These experiments show that our approach achieves performance comparable to fully supervised approaches and outperforms other weakly supervised learning baselines; (iv) we release an open-source MATLAB implementation of our algorithm as part of the DOGMA library [34].

The rest of this paper is organized as follows. We review related works in sec. 2. Sec. 3 defines the problem of learning from images with captions and casts it into the candidate labeling sets framework. Sec. 4 presents the Maximum Margin Set (MMS) algorithm, which solves the learning problem defined before. We then report experiments on artificial datasets in sec. 5. We also apply our framework to two real-world tasks: learning face classifiers from news items (sec. 6) and learning both face and action (body pose) classifiers jointly (sec. 7). We show that learning both at the same time reduces the underlying corresponding ambiguity, and solves the name-to-face and verb-to-body assignment problems better than when tackling either task alone. We conclude the paper with discussions and possible extensions (sec. 8).

2 Related Works

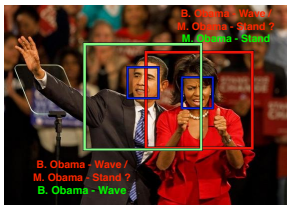
Images and Captions/Tags: Learning visual classifiers from images with tags has been a very active line of research in recent years [3], [24], [41]. These approaches must resolve the correspondence between image segments and tags, which are typically nouns (e.g. tiger, grass, car). Because the tags are manually annotated to be descriptive for the image, algorithms can safely assume that nearly all tags should correspond to one or more image regions.

The problem of naming faces in images and videos using natural text sources has been particularly well studied [4], [10], [16], [26]. These works exploit the fact that often the names of the persons in the image are mentioned in the caption. Therefore, a caption contains possible labels for the faces in the corresponding image. However, an imaged person might not be mentioned in the caption and vice-versa. Hence, the level of noise and ambiguity in natural captions is typically higher compared to image tags. Various kinds of task-specific knowledge has also been integrated to improve learning performance, such as that two faces in one image can not be associated with the same name [4], or exploiting the motion of the mouth and the gender of a person [10].

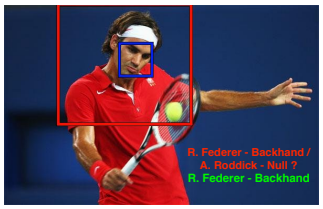
Recently, a few works went beyond modeling a single type of word, and start to exploit the structure of sentences in the caption. Gupta and David [27] model prepositions in addition to nouns (e.g. 'bear in water', 'car on street'). This prunes down the space of possible labelings. In Jie et al. [29], we model both names and action verbs jointly, and show that face and pose information help each other by reducing the correspondence ambiguity.

Related learning frameworks: Our problem is different from semi-supervised learning [46], where the learner has access to a set of labeled examples as well as a set of unlabeled examples. Instead, it is closer to the ambiguously labeled leaning or partially labeled learning setting [10], [23], [28], [31], where each training example is associated with multiple labels, only one of which is correct. Many approaches to such problems use the EM algorithm to estimate model parameters and the correct labels [23], [31]. The recent work of [10] is the most related to this paper, as it proposes a convex learning formulation based on minimizing an ambiguous loss function. In this paper, we generalize the ambiguous function to the multiple instances case, and use a non-convex learning formulation which achieves better performance than the convex learning formulation (sec. 4.5).

Our work is also related to multi-label learning (MLL) [6], where each example is assigned multiple labels, any subset of which can be correct. Other related lines of research are multi-instance learning (MIL) [2], [14], and multi-instance multi-label learning (MIML) [44], [45]. MIML extends the two-label MIL setup to multiple labels. In both setups, instances are grouped into bags. The labels of the individual instances are not given. Instead, labels are given to the bags. However, contrary to our framework, in MIML noisy labels are not allowed: all the given labels for a bag are



*US Democratic presidential candidate Senator **Barack Obama** waves to supporters together with his wife **Michelle Obama** standing beside him at his North Carolina and Indiana primary election night rally in Raleigh.*



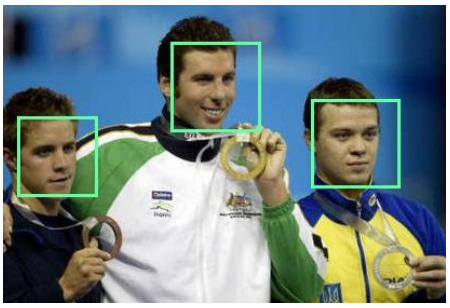
*Four sets ... **Roger Federer** prepares to **hit a backhand** in a quarter-final match with **Andy Roddick** at the US Open.*

$$\mathcal{X} = \left\{ \boldsymbol{x} \right\} = \left\{ \left[\boldsymbol{x}^{(1)} \;,\; \boldsymbol{x}^{(2)} \right] \right\}$$

$$\mathcal{Z} = \left\{ \begin{matrix} \boldsymbol{Z}_1 \\ \boldsymbol{Z}_2 \end{matrix} \right\} = \left\{ \begin{matrix} \left[z_1^{(1)} \text{:Federer}, \; z_1^{(2)} \text{:Backhand} \right] \\ \left[z_2^{(1)} \text{:Roddick}, \quad z_2^{(2)} \text{:Null} \quad \right] \end{matrix} \right\}$$

$$\boldsymbol{Y} = \left\{ \left[y^{(1)} \text{:Federer}, y^{(2)} \text{:Backhand} \right] \right\} \quad \textbf{(Unknown)}$$

Fig. 1. (Left, Middle) Two examples of image-caption pairs for the "who is doing what" task [29]. The face and upper body of the persons in the image are marked by bounding-boxes. We stress that a caption might contain names and/or verbs not visible in the image, and vice-versa. (Right) our candidate labeling set notation for the example in the middle. The image $\mathcal{X}$ contains one region $\boldsymbol{x}$, which has two attributes: the person name and the verb describing the action he is performing. The candidate labeling set $\mathcal{Z}$ contains two candidate labeling vectors $\boldsymbol{z}_1$ and $\boldsymbol{z}_2$. Each labeling vector encodes one label for every attribute of the region. Importantly, note how [Roddick, Backhand] is not a candidate, as Roddick is not the subject of the verb "hit a backhand" in the caption. The true labeling vector $\boldsymbol{Y}$ is unknown and must be recovered by the algorithm.



*Australia's gold medalist **Grant Hackett** (C), Ukraine's silver medalist **Igor Chervynsky** and USA's bronze medalist **Erik Vendt** show their medals following the 1500 metres freestyle race at the 10th World Swimming Championships in Barcelona July 27, 2003. Hackett clocked fourteen minutes 43.14 seconds.*

Fig. 2. Example of an image-caption pair containing spatial indicators. The spatial indicator (C) indicates **Hackett** is the person in the middle, reducing the ambiguity in labels assignment.

correct. Moreover, current MIL and MIML algorithms usually rely on a 'key' instance in the bag [2] or they transform each bag into a single-instance representation [45]. Instead, our algorithm makes an explicit effort to label every instance in a bag and to consider all of them during learning.

Latent Structure SVMs: Our algorithm is also related to Latent Structural SVMs [18], [42], where the correct labels are considered as latent variables. Wang and Mori [41] recently proposed a discriminative latent model for annotating scene images given object nouns as tags (e.g. tiger, grass). They model the ground-truth region-to-annotation mapping and the overall scene label as latent variables.

3 Problem Definition

In this section, we define the problem of learning from images with captions, and establish the notation that will be used in the rest of the paper. We denote vectors by bold letters, e.g. $\boldsymbol{x}, \boldsymbol{y}$, and use calligraphic letters for sets, e.g. $\mathcal{X}$. Fig. 1 (right) gives an example of our setup. In sec. 6 and sec. 7 we will give several examples on how to cast existing problems into our framework.

Input data: The input is a collection of N image and caption pairs $\{\mathcal{X}_i, \mathcal{C}_i\}_{i=1}^{N}$. An image $\mathcal{X}_i$ consists of M_i regions $\mathcal{X}_i = \{\boldsymbol{x}_{i,m}\}_{m=1}^{M_i}$, and $\boldsymbol{x}_{i,m} \in \mathbb{R}^d$. Each image has an associated caption $\mathcal{C}_i$, which implicitly provides partial labels for the image. Many real-world objects can belong to multiple concepts simultaneously. For example, an image region can bear several attributes: red (color), metal (texture) and car (object category). Quite often these attributes are correlated, so we argue that it is useful to model them together, using a label for each attribute. Without loss of generality, we assume that labels $\boldsymbol{Y}_i = \{\boldsymbol{y}_{i,m}\}_{m=1}^{M_i}$ exist for every image region, but they are unknown during training. We consider them as latent variables. The latent variables $\{\boldsymbol{y}_{i,m}\}_{m=1}^{M_i}$ encode the labels for each region in the images. Each $\boldsymbol{y}_{i,m}$ is either a set of labels $\{y_{i,m}^{(p)}\}_{p=1}^{P}$ or a single label $y_{i,m}$ (i.e. $P = 1$), where P is the total number of attributes we model simultaneously. Each label $y_{i,m}^{(p)} \in \mathbb{Y}^{(p)} := \{1, 2, \cdots, K^{(p)}\}$ indicates a specific attribute of a region, and $K^{(p)}$ denotes the number of possible different labels for the attribute p.

Candidate Labeling Sets: Our goal is to learn from the input image-caption pairs a classification function $f : \boldsymbol{x} \rightarrow \boldsymbol{y}$ to classify regions of a new test image. The caption for the test image, when available, could still be used as an extra source of information to guide the prediction, but it is not required. Although the true labels of a training image are unknown, the accompanying caption usually describes the image. We assume that the labels of the regions only come from the caption. The learning algorithm should label with *null* any region whose true label is not mentioned in the caption. A label corresponds to a word, or to a few words with the same meaning (e.g. "Barack Obama" and "President of the

USA"). In the rest of the paper we will only use the term "word". Based on these assumptions, we generate the set of all possible assignments of the words in a caption to the regions in the corresponding image i, which we call Candidate Labeling Set (CLS) $\mathcal{Z}_i$. We use L_i to denote the number of candidate assignments in $\mathcal{Z}_i = \{\boldsymbol{Z}_{i,l}\}_{l=1}^{L_i}$, with each $\boldsymbol{Z}_{i,l} \in R^{P \times M_i}$. In other words, there are L_i different possible combinations of labels for the regions in the image i. Only one of these candidate assignments is the true labeling, while the others are only partially correct or even completely wrong. Note that this is *not* equivalent to simply associating L_i candidate labels independently to each region. Instead, our definition explicitly encodes the constrains between multiple regions and labels. To clarify this point, consider a simple example where we have two regions $\{\boldsymbol{x}_{i,1}, \boldsymbol{x}_{i,2}\}$ with two attributes each (color and object category). If it is known that they can only come from classes "red-car" or "blue-motor", and that no two regions can have the same label, then $\boldsymbol{z}_{i,1} = [\text{red-car}, \text{blue-motor}], \boldsymbol{z}_{i,2} = [\text{blue-motor}, \text{red-car}]$ will be the Candidate Labeling Vectors (CLVs) for this bag. Other possibilities such as $[\text{blue-car}, \text{red-motor}]$, $[\text{red-car}, \text{red-car}]$ are excluded. Another example could be an image with three regions, each of which could be labeled either "chair" or "elephant". However, we know there cannot be both chairs and elephants in the same image. Such a structure can be encoded in our CLSs, but not in simple independent label sets for each region.

Constraints between words: As the size of the regions and the number of words grow, the number of admissible labelings becomes intractable. To keep the problem tractable, we could first filter out uninteresting words such as interjections and conjunctions, and maintain a dictionary with only the words we want to model. Each of these words will correspond to a different label. However, the number of admissible labelings can still be very large after the filtering. Let W_i be the number of modeled words in the caption of an image i with M_i regions. In the most general case, this image has $L_i = W_i^{M_i}$ admissible labelings even when only one label can be assigned to each region. In this unconstrained scenario, the supervision information from the caption is very low for large W_i. Fortunately, captions frequently contain valuable context cues which we can extract using NLP tools [1], [13]. These context cues can be translated into constraints to remove assignments from $\mathcal{Z}_i$. In addition, we can also reduce the size of $\mathcal{Z}_i$ by incorporating high-level domain knowledge. As L_i decreases when more constraints are added, the CLS $\mathcal{Z}_i$ becomes less ambiguous. This scenario allows us to design interesting learning algorithms, which we present in the next Section.

Our CLSs framework supports several types of useful constraints:

- [C1] *Word type matches region type*. For example, a name can only be associated with a face region, and/or a verb can only be associated with a person body region (where such regions are detected beforehand, e.g. by an off-the-shelf face detector [38], [35] or an upper-body detector [21]). This type of constraint has been used in several works [4], [25], [29] (see fig. 1 for example).
- [C2] *Sentence structure*. Multiple words grammatically connected in the caption must be assigned to spatially related image regions or even to the same region. Several kinds of connections in the caption can be used to eliminate labelings from $\mathcal{Z}_i$ which violate the resulting constraints: (a) noun-adjective [39], e.g. a "red car", where both "red" and "car" are attributes of the same region; (b) name-verb [29], e.g. "Roger Federer hits a backhand", "Roger Federer" is the subject of "hits backhand" and therefore point to the same person in the image. Therefore, any labeling that assigns "Roger Federer" to a certain face region, must assign "hits backhand" to the body region of the *same person*; (c) noun-preposition-noun [27], e.g. "sun in the sky" indicates that the two regions are close to each other, and the "sun" region is surrounded by the "sky" region. In general, this kind of connection conveys information about the spatial relationship between two regions.
- [C3] *Uniqueness*. Some words can only appear once in the image. Two face regions in the same image cannot be associated to the same name [4], [25], [29].
- [C4] *Spatial indicators*. Captions sometimes contain spatial position indicators [4] such as "(L)" and "left". These suggest the relative spatial position of an image region w.r.t. the others. An example of this kind of connection is shown in fig. 2. The noun-preposition-noun structure discussed in [C2] can also be considered as a spatial indicator.

Based on these constraints, we can explicitly enumerate the set of admissible assignments $\mathcal{Z}_i$ from the caption $\mathcal{C}_i$ in several interesting problems. Hence, we replace the captions by the CLSs $\{\mathcal{Z}_i\}_{i=1}^N$. In a few other cases, memory limitations prevents us from explicitly storing all assignments $\mathcal{Z}_i$. In such a case we store the words and the constraints, which we can use to generate subsets of $\mathcal{Z}_i$ "on the fly" during learning.

In this setting, the training data are provided in the form $\{\mathcal{X}_i, \mathcal{Z}_i\}_{i=1}^N$. Each image $\mathcal{X}_i$ is associated with a set of CLVs $\mathcal{Z}_i$ (including one which is fully correct). Thus, our goal is to design a learning algorithm which learns classifiers from input data in this special form. Along the way to learning these classifiers, our algorithm also selects one CLV for each image, thus resolving the correspondence between image regions and words in the caption.

4 Learning from Candidate Labeling Sets and the MMS Algorithm

In sec. 3 we have discussed how to transform the problem of learning from image-caption pairs into the CLS

problem. In this section, we first propose a large margin formulation of the CLS problem (sec. 4.1 to 4.4), then we present an efficient algorithm to optimize the proposed formulation (sec. 4.5 and 4.6).

Let $\mathcal{X}$ be the generic bag with M instances $\{\boldsymbol{x}_1, \ldots, \boldsymbol{x}_m, \ldots, \boldsymbol{x}_M\}$, $\mathcal{Z} = \{\boldsymbol{Z}_1, \ldots, \boldsymbol{Z}_l, \ldots, \boldsymbol{Z}_L\}$ the generic set of CLVs. Under this representation, $\mathcal{X}$ can be an image with M regions, and an instance $\boldsymbol{x}_m$ is a vector of appearance features describing the m-th region. An image region can either be a rectangular bounding box (e.g. a face [38]) found by an object detector, or an arbitrarily shaped segment found by an unsupervised segmentation algorithm [19]. Furthermore, let $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}, \boldsymbol{Z} = \{\boldsymbol{z}_1, \ldots, \boldsymbol{z}_M\}$ be two labeling vectors, where each element $\boldsymbol{z}, \boldsymbol{y} \in \mathbb{Y}^P$ is a label, with P denoted the number of attributes we model for each instance. We also assume a uniform prior over the CLVs in the CLSs, i.e. $p(\boldsymbol{Z}_i) = p(\boldsymbol{Z}_j),\ \forall\ \boldsymbol{Z}_i, \boldsymbol{Z}_j \in \mathcal{Z}$. Later, we will discuss the possibility to extend these probabilities when the priors for each $\boldsymbol{Z}_i$ are known.

4.1 Prediction functions

Given the training data $\{\mathcal{X}_i, \mathcal{Z}_i\}_{i=1}^N$, we want to learn a linear prediction function which can work on individual instances. Motivated by the linear model in structural SVMs [32], we define the score functions as

$$s_{\boldsymbol{w}}(\boldsymbol{x}, y^{(p)}) = \boldsymbol{w} \cdot \phi(\boldsymbol{x}, y^{(p)}) \ ,$$

where $\phi(\cdot, \cdot)$ is the feature mapping function, which creates a joint feature vector describing the relationship between the original input vector $\boldsymbol{x}$ and the label $y^{(p)}$, with $\boldsymbol{w}$ being the classification hyperplane. Intuitively, the score function quantifies how confident the model is to assign instance $\boldsymbol{x}$ to class $y^{(p)}$, and the predicted label is the one with the highest score: $\arg\max_{y^{(p)} \in \mathbb{Y}^{(p)}}\ s_{\boldsymbol{w}}(\boldsymbol{x}, y^{(p)})$.

We also define the linear prediction function for an image region $\boldsymbol{x}$ as

$$\begin{aligned} f_{\boldsymbol{w}}(\boldsymbol{x}) =& \operatorname*{argmax}_{y^{(p)} \in \mathbb{Y}^{(p)}} \sum_{p=1}^{P} s_{\boldsymbol{w}}(\boldsymbol{x}, y^{(p)}) \\ =& \operatorname*{argmax}_{\boldsymbol{y} \in \mathbb{Y}} \boldsymbol{w} \cdot \psi(\boldsymbol{x}, \boldsymbol{y}) \ , \end{aligned}$$

where $\psi(\boldsymbol{x}, \boldsymbol{y}) = \sum_{p=1}^{P} \phi(\boldsymbol{x}, y^{(p)})$ is a joint feature mapping vector between the region $\boldsymbol{x}$ and the labeling vector $\boldsymbol{y}$. This definition includes the special case of training different hyperplanes, one for each class. Indeed $\psi(\boldsymbol{x}, \boldsymbol{y})$ can be defined as

$$\psi(\boldsymbol{x}, \boldsymbol{y}) = [\overbrace{\mathbf{0},\ \cdots,\ \mathbf{0},\ \underbrace{\phi^{(1)}(\boldsymbol{x})}_{y^{(1)}\text{-th}},\ \mathbf{0},\ \cdots,\ \mathbf{0}}^{K^{(1)}},\ \cdots,$$
$$\overbrace{\mathbf{0},\ \cdots,\ \mathbf{0},\ \underbrace{\phi^{(p)}(\boldsymbol{x})}_{y^{(p)}\text{-th}},\ \mathbf{0},\ \cdots,\ \mathbf{0}}^{K^{(p)}},\ \cdots] \ ,$$

where $\phi^{(p)}(\cdot)$ is a transformation that depends only on the data and the attribute p. In this case the classifier $\boldsymbol{w}$ is parameterized by $\sum_{p=1}^{P} K^{(p)}$ hyperplanes $\boldsymbol{w}_{y^{(p)}}$.

We can now define the score function for the image as $\mathbf{S}_{\boldsymbol{w}}(\mathcal{X}, \boldsymbol{Y})$, which intuitively is gathering from each region in $\mathcal{X}$ the confidence on the labels encoded in $\boldsymbol{Y}$. With the definitions above, we define the function $\mathbf{S}$ as

$$\begin{aligned} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}, \boldsymbol{Y}) &= \sum_{m=1}^{M} s_{\boldsymbol{w}}(\boldsymbol{x}_m, \boldsymbol{y}_m) \\ &= \sum_{m=1}^{M} \boldsymbol{w} \cdot \psi(\boldsymbol{x}_m, \boldsymbol{y}_m) \\ &= \boldsymbol{w} \cdot \Phi(\mathcal{X}, \boldsymbol{Y}) \ , \end{aligned} \tag{1}$$

where we have $\Phi(\mathcal{X}, \boldsymbol{Y}) = \sum_{m=1}^{M} \psi(\boldsymbol{x}_m, \boldsymbol{y}_m)$. Given the CLS $\mathcal{Z}$, the predictions of the classifier are computed as $\mathbf{F}_{\boldsymbol{w}}(\mathcal{X}, \mathcal{Z}) = \arg\max_{\boldsymbol{Z} \in \mathcal{Z}}\ \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}, \boldsymbol{Z})$.

Remark 1: If the prior probabilities of the CLVs $\boldsymbol{z}_l \in \mathcal{Z}$ are also available, they can be incorporated into the score function by slightly modifying the feature mapping function in eq. (1) to $p(\boldsymbol{Z}_i) \cdot \Phi(\mathcal{X}, \boldsymbol{Z}_i)$, where each $p(\boldsymbol{Z}_i)$ is the prior probability for $\boldsymbol{Z}_i \in \mathcal{Z}$.

4.2 Ambiguous loss functions

In the supervised learning setup, many loss functions have been proposed based on minimization of a convex upper bound of an arbitrary risk measurement function $\boldsymbol{\Delta} : \mathbb{Y} \times \mathbb{Y} \rightarrow R$, which quantifies how much a predicted label differs from the true label. A classic loss function is the 0/1 loss:

$$\boldsymbol{\Delta}_{01}(\boldsymbol{Z}, \boldsymbol{Y}) = \sum_{m=1}^{M} \sum_{p=1}^{P} \mathbf{1}(z_m^{(p)} \neq y_m^{(p)}) \ ,$$

where $\mathbf{1}(\cdot)$ is the indicator function, $\boldsymbol{Y} = \{\boldsymbol{y}_1, \ldots, \boldsymbol{y}_M\}$ are the true labels of regions $\boldsymbol{x}_m$, and $\boldsymbol{Z}$ are the predicted labels. Hence, $\boldsymbol{\Delta}_{01}(\boldsymbol{Z}, \boldsymbol{Y})$ simply counts the number of mislabeled attributes over all regions.

However, in our setup the true labeling is unknown, and we only have access to the CLS $\mathcal{Z}$, knowing that the true labeling vector is in it. So we propose to use an ambiguous version of the loss $\boldsymbol{\Delta}_{01}$, as a proxy for it:

$$\boldsymbol{\Delta}_A(\boldsymbol{Z}, \mathcal{Z}) = \min_{\boldsymbol{Z}' \in \mathcal{Z}} \boldsymbol{\Delta}_{01}(\boldsymbol{Z}, \boldsymbol{Z}') \ .$$

This loss function underestimates the true loss, while our goal is to minimize the true loss. Nevertheless, we can prove a strong connection between the ambiguous loss $\boldsymbol{\Delta}_A(\boldsymbol{Z}, \mathcal{Z})$ and the true loss $\boldsymbol{\Delta}_{01}(\boldsymbol{Z}, \boldsymbol{Y})$. The following proposition shows that, in expectation, the ambiguous loss upper bounds the true 0/1 loss up to a constant multiplicative factor. To prove this, we use the theorems stated in [10, Proposition 3.1 to 3.3], and define an *ambiguity degree* factor η for a region $\boldsymbol{x}_m$. The value of η corresponds to the maximum probability of a noise label (i.e. $\forall y \in \mathbb{Y}^{(p)} \setminus y_m^{(p)}$) co-occurring with a true label $y_m^{(p)}$

in the CLS $\mathcal{Z}$, over all labels and examples generated by an unknown distribution.

Proposition 1: $E\left[\boldsymbol{\Delta}_{01}(\boldsymbol{Z},\boldsymbol{Y})\right] \leq \frac{1}{1-\eta} E\left[\boldsymbol{\Delta}_A(\boldsymbol{Z},\mathcal{Z})\right]$

Proof [Sketch]: Define

$$\Delta_A(z_m^{(p)},\mathcal{Z}_m^{(p)}) = \min_{z'\in\mathcal{Z}_m^{(p)}} \Delta_{01}(z_m^{(p)},z') = \mathbf{1}(z_m^{(p)} \notin \mathcal{Z}_m^{(p)})$$

the ambiguous loss for a single region $\boldsymbol{x}_m$ and an attribute p, with $\mathcal{Z}_m^{(p)}$ being the corresponding CLS for the region. So we can use Proposition 3.1 of [10] to obtain

$$E[\Delta_{01}(z_m^{(p)},y_m^{(p)})] \leq \frac{1}{1-\eta} E[\Delta_A(z_m^{(p)},\mathcal{Z}_m^{(p)})] \ .$$

Using the definition of the true $0/1$ loss and the linearity of expectation, and summing over m and p, we have

$$E\left[\boldsymbol{\Delta}_{01}(\boldsymbol{Z},\boldsymbol{Y})\right] \leq \frac{1}{1-\eta} E\left[\sum_{m=1}^{M}\sum_{p=1}^{P} \mathbf{1}(z_m^{(p)} \notin \mathcal{Z}_m^{(p)})\right] \ ,$$

while using the relationship that $\bigcup_{m,p} \mathcal{Z}_m^{(p)} \supseteq \mathcal{Z}$, we obtain

$$E\left[\sum_{m=1}^{M}\sum_{p=1}^{P} \mathbf{1}(z_m^{(p)} \notin \mathcal{Z}_m^{(p)})\right] \leq E\left[\boldsymbol{\Delta}_A(\boldsymbol{Z},\mathcal{Z})\right] \ .$$

Combining them results in Proposition 1. □

Remark 2: The tightness of the above bound directly relates to the ambiguity degree η. When $\eta = 0$, we have only one labeling vector in every labeling set, i.e. $L_i = 1$. In this case, the problem becomes standard supervised learning, and the bound is tight. On the other extreme, when $\eta = 1$, which means a certain noise label always co-occurs with a true label $y^{(p)}$, it is impossible to distinguish them. One weakness of the stated bound is that it becomes very loose when there is a noise label that makes η very large, because that η equals the maximum probability of co-occurring among all the possible noise labels, although it only affects a few true labels. Nevertheless, using the extensions of Proposition 3.2 and Proposition 3.3 in [10], it is possible to obtain label-specific bounds, which enable to retain good learning performance on the subset of labels with low label-specific ambiguity degrees.

Hence, by minimizing the ambiguous loss we are actually minimizing an upper bound of the expected true loss. It is known that direct minimization of this loss is hard [12]. Therefore, in the following we introduce another loss that upper bounds $\boldsymbol{\Delta}_A$ which can be minimized efficiently:

$$\begin{aligned}\ell_A(\mathcal{X},\mathcal{Z};\boldsymbol{w}) = |&\max_{\bar{\boldsymbol{Z}}\notin\mathcal{Z}} \left(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}},\mathcal{Z}) + \mathbf{S}_{\boldsymbol{w}}(\mathcal{X},\bar{\boldsymbol{Z}})\right) \\ &- \max_{\boldsymbol{Z}\in\mathcal{Z}} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X},\boldsymbol{Z})|_+ \ , \end{aligned} \tag{2}$$

where $|x|_+ = \max(0,x)$. The following proposition shows that ℓ_A upper bounds $\boldsymbol{\Delta}_A$.

Proposition 2: $\ell_A(\mathcal{X},\mathcal{Z};\boldsymbol{w}) \geq \boldsymbol{\Delta}_A(\mathcal{X},\mathcal{Z};\boldsymbol{w})$.

Proof : Define $\hat{\boldsymbol{z}} = \arg\max_{\boldsymbol{z}\in\mathcal{Y}^M} \mathbf{S}(\mathcal{X},\boldsymbol{z};\boldsymbol{w})$. If $\hat{\boldsymbol{Z}} \in \mathcal{Z}$ then $\ell_A(\mathcal{X},\mathcal{Z};\boldsymbol{w}) \geq \boldsymbol{\Delta}_A(\mathcal{X},\mathcal{Z};\boldsymbol{w}) = 0$. We now consider the case in which $\hat{\boldsymbol{Z}} \notin \mathcal{Z}$. We have that

$$\begin{aligned}&\boldsymbol{\Delta}_A(\mathcal{X},\mathcal{Z};\boldsymbol{w}) \\ &\leq \boldsymbol{\Delta}_A(\hat{\boldsymbol{Z}},\mathcal{Z}) + \mathbf{S}_{\boldsymbol{w}}(\mathcal{X},\hat{\boldsymbol{Z}}) - \max_{\boldsymbol{z}\in\mathcal{Z}} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X},\boldsymbol{Z}) \\ &\leq \max_{\bar{\boldsymbol{Z}}\notin\mathcal{Z}} \left(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}},\mathcal{Z}) + \mathbf{S}_{\boldsymbol{w}}(\mathcal{X},\bar{\boldsymbol{Z}})\right) - \max_{\boldsymbol{Z}\in\mathcal{Z}} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X},\boldsymbol{Z}) \\ &\leq \ell_A(\mathcal{X},\mathcal{Z};\boldsymbol{w}) \ . \end{aligned}$$ □

4.3 A probabilistic interpretation

It is possible to gain an additional intuition on the proposed loss function ℓ_A through a probabilistic interpretation of the problem. It is helpful to look at the discriminative model for supervised learning first, where the goal is to learn the model parameters θ for the function $P(y|\boldsymbol{x};\theta)$, from a pre-defined modeling class Θ. Instead of directly maximizing the log-likelihood for the training data, an alternative way is to maximize the log-likelihood ratio between the correct label and the most likely incorrect one [11]. On the other hand, in the CLS setting the correct labeling vector for $\mathcal{X}$ is unknown, but it is known to be a member of the candidate set $\mathcal{Z}$. Hence we could maximize the log-likelihood ratio between $P(\mathcal{Z}|\mathcal{X};\theta)$ and the most likely incorrect labeling vector which is not a member of $\mathcal{Z}$ (denoted as $\bar{\boldsymbol{z}}$). However, the relation between different vectors in $\mathcal{Z}$ are not known, so the inference could be arbitrarily hard. Instead, we could approximate the problem by considering just the most likely correct member of $\mathcal{Z}$. It can be easily verified that $\max_{\boldsymbol{Z}\in\mathcal{Z}} P(\boldsymbol{Z}|\mathcal{X};\theta)$ is a lower bound of $P(\mathcal{Z}|\mathcal{X};\theta)$, as $\boldsymbol{Z} \subseteq \mathcal{Z}$. Hence the learning problem becomes that of minimizing the ratio for the bag:

$$-\log \frac{P(\mathcal{Z}|\mathcal{X};\theta)}{\max_{\bar{\boldsymbol{Z}}\notin\mathcal{Z}} P(\bar{\boldsymbol{Z}}|\mathcal{X};\theta)} \approx -\log \frac{\max_{\boldsymbol{Z}\in\mathcal{Z}} P(\boldsymbol{Z}|\mathcal{X};\theta)}{\max_{\bar{\boldsymbol{Z}}\notin\mathcal{Z}} P(\bar{\boldsymbol{Z}}|\mathcal{X};\theta)} \ . \tag{3}$$

If we assume independence between the instances in the bag and different attributes of an instance, eq. (3) can be factorized as:

$$\begin{aligned}&-\log \frac{\max_{\boldsymbol{Z}\in\mathcal{Z}} \prod_{m=1}^{M}\prod_{p=1}^{P} P(z_m^{(p)}|\boldsymbol{x}_m;\theta)}{\max_{\bar{\boldsymbol{Z}}\notin\mathcal{Z}} \prod_{m=1}^{M}\prod_{p=1}^{P} P(\bar{z}_m^{(p)}|\boldsymbol{x}_m;\theta)} \\ =&\max_{\bar{\boldsymbol{Z}}\notin\mathcal{Z}} \sum_{m=1}^{M}\sum_{p=1}^{P} \log P(\bar{z}_m^{(p)}|\boldsymbol{x}_m;\theta) \\ &- \max_{\boldsymbol{Z}\in\mathcal{Z}} \sum_{m=1}^{M}\sum_{p=1}^{P} \log P(z_m^{(p)}|\boldsymbol{x}_m;\theta) \ . \end{aligned}$$

If we take the margin into account, and assume a linear model for the log-posterior-likelihood, we obtain the loss function in eq. (2).

Inferences are usually intractable for large bags if we consider different type of constrains explicitly in other algorithms. However, these constrains can still be taken into account in CLS setup as they are encoded in $\mathcal{Z}$.

4.4 Maximum Margin Set (MMS)

Using the square norm regularizer as in the SVM and the loss function (2), we have the following optimization problem:

$$\min_{\boldsymbol{w}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} \ell_A\left(\mathcal{X}_i, \mathcal{Z}_i; \boldsymbol{w}\right) \ . \tag{4}$$

This optimization problem is non-convex due to the second $\max(\cdot)$ inside the loss (2). To convexify this problem, one could approximate the second $\max(\cdot)$ with the average over all labeling vectors in $\mathcal{Z}_i$. Similar strategies have been used in analogous problems [10], [44]. However, the approximation could be very loose if the number of labeling vectors is large. Fortunately, although the loss function is not convex, a good local minimum can be found using the constrained concave-convex procedure (CCCP) [37], [43].

4.5 Optimizing the MMS problem with CCCP

To optimize (2) using CCCP, we first rewrite it as

$$\begin{aligned}
\min_{\boldsymbol{w}} \quad & \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} \xi_i \\
\text{s.t.} \quad & \max_{\bar{\boldsymbol{Z}} \notin \mathcal{Z}_i} \left(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}}, \mathcal{Z}_i) + \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \bar{\boldsymbol{Z}})\right) \\
& - \max_{\boldsymbol{Z} \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \boldsymbol{Z}) \leq \xi_i, \quad i = 1, \ \ldots, \ N \\
& \xi_i \geq 0, \quad i = 1, \ \ldots, \ N \ .
\end{aligned}$$

In this formulation, the objective function is convex, while the first set of constraints can be written as the difference of a convex function and a concave function. The CCCP solves the optimization problem using an iterative minimization process. At each round r, given an initial $\boldsymbol{w}^{(r)}$, the CCCP replaces the concave part of the constraints with its first-order Taylor expansion at $\boldsymbol{w}^{(r)}$, and then sets $\boldsymbol{w}^{(r+1)}$ to the solution of the relaxed constrained optimization problem. When this function is non-smooth, such as $\max_{\boldsymbol{z} \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \boldsymbol{Z})$ in our formulation, the gradient in the Taylor expansion must be replaced by the subgradient[1]. Thus, at the r-th round, the CCCP replaces $\max_{\boldsymbol{Z} \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \boldsymbol{Z})$ by

$$\max_{\boldsymbol{Z} \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}^{(r)}}(\mathcal{X}_i, \boldsymbol{Z}) + (\boldsymbol{w} - \boldsymbol{w}^{(r)}) \cdot \partial \left(\max_{\boldsymbol{Z} \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \boldsymbol{Z}) \right) \ . \tag{5}$$

The subgradient of a point-wise maximum function $g(\boldsymbol{x}) = \max_i g_i(\boldsymbol{x})$ is the convex hull of the union of subdifferentials of the subset of the functions $g_i(\boldsymbol{x})$ which equal $g(\boldsymbol{x})$ [5]. Defining by $\mathcal{C}_i^{(r)} = \{\boldsymbol{Z} \in \mathcal{Z}_i : \mathbf{S}_{\boldsymbol{w}^{(r)}}(\mathcal{X}_i, \boldsymbol{Z}) = \max_{\boldsymbol{z}' \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}^{(r)}}(\mathcal{X}_i, \boldsymbol{Z})\}$, the subgradient of the function $\max_{\boldsymbol{Z} \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \boldsymbol{Z})$ equals to $\sum_l \alpha_{i,l}^{(r)} \partial \mathbf{S}_{\boldsymbol{w}}(\mathcal{X}_i, \boldsymbol{Z}_{i,l}) = \sum_l \alpha_{i,l}^{(r)} \Phi(\mathcal{X}_i, \boldsymbol{Z}_{i,l})$, with $\sum_l \alpha_{i,l}^{(r)} = 1$, and $\alpha_{i,l}^{(r)} \geq 0$ if $\boldsymbol{Z}_{i,l} \in \mathcal{C}_i^{(r)}$ and $\alpha_{i,l}^{(r)} = 0$ otherwise. Hence we have

$$\begin{aligned}
& \sum_l \alpha_{i,l}^{(r)} \boldsymbol{w}^{(r)} \cdot \Phi(\mathcal{X}_i, \boldsymbol{Z}_{i,l}) \\
&= \max_{\boldsymbol{Z} \in \mathcal{Z}_i} \left(\boldsymbol{w}^{(r)} \cdot \Phi(\mathcal{X}_i, \boldsymbol{Z}) \right) \sum_{l : \boldsymbol{Z}_{i,l} \in \mathcal{C}_i^{(r)}} \alpha_{i,l}^{(r)} \\
&= \max_{\boldsymbol{Z} \in \mathcal{Z}_i} \left(\boldsymbol{w}^{(r)} \cdot \Phi(\mathcal{X}_i, \boldsymbol{Z}) \right) \ .
\end{aligned}$$

Combining this with (5), the constraints become

$$\begin{aligned}
& \max_{\bar{\boldsymbol{Z}} \notin \mathcal{Z}_i} \left(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}}, \mathcal{Z}_i) + \boldsymbol{w} \cdot \Phi(\mathcal{X}_i, \bar{\boldsymbol{Z}}) \right) \\
& \qquad - \boldsymbol{w} \cdot \sum_{\boldsymbol{Z}_{i,l} \in \mathcal{C}_i^{(r)}} \alpha_{i,l}^{(r)} \Phi(\mathcal{X}_i, \boldsymbol{Z}_{i,l}) \leq \xi_i \ .
\end{aligned}$$

Hence the relaxed convex optimization program at the r-th round of the CCCP is equivalent to the problem

$$\min_{\boldsymbol{w}} \quad \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{cccp}}^{(r)}\left(\mathcal{X}_i, \mathcal{Z}_i; \boldsymbol{w}\right), \tag{6}$$

where

$$\begin{aligned}
\ell_{\text{cccp}}^{(r)}\left(\mathcal{X}_i, \mathcal{Z}_i; \boldsymbol{w}\right) = \big| & \max_{\bar{\boldsymbol{Z}} \notin \mathcal{Z}_i} \left(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}}, \mathcal{Z}_i) + \boldsymbol{w} \cdot \Phi(\mathcal{X}_i, \bar{\boldsymbol{Z}}) \right) \\
& - \boldsymbol{w} \cdot \sum_{\boldsymbol{Z}_{i,l} \in \mathcal{C}_i^{(r)}} \alpha_{i,l}^{(r)} \Phi(\mathcal{X}_i, \boldsymbol{Z}_{i,l}) \big|_+ \ .
\end{aligned}$$

The procedure of the CCCP algorithm is outlined in Algorithm 1. It is guaranteed to decrease the objective function and it converges to a local minimum solution of problem (4) [37], [43].

We are free to choose the values of the $\alpha_{i,l}^{(r)}$ in the convex hull. Since the algorithm is susceptible to a local minima, its performance could possibly be sensitive to initialization. Here we choose to set $\alpha_{i,l}^{(r)} = 1/|\mathcal{C}_i^{(r)}|$ for $\forall \boldsymbol{z}_{i,l} \in \mathcal{C}_i^{(r)}$. With our choice of $\alpha_{i,l}^{(r)}$, in the first round of the CCCP when $\boldsymbol{w}$ is initialized at $\mathbf{0}$, the second $\max(\cdot)$ in (2) is approximated by the average over all the labeling vectors. In this way, the first round of the algorithm is similar to the convex relaxation methods in [10], [44], but here the later iterations will improve the solution.

Algorithm 1 The CCCP algorithm for solving MMS

1: **initialize:** $\boldsymbol{w}^{(1)} = \mathbf{0}$
2: **repeat**
3: Set $\mathcal{C}_i^{(r)} = \{\boldsymbol{Z} \in \mathcal{Z}_i : \mathbf{S}_{\boldsymbol{w}^{(r)}}(\mathcal{X}_i, \boldsymbol{Z}) = \max_{\boldsymbol{Z}' \in \mathcal{Z}_i} \mathbf{S}_{\boldsymbol{w}^{(r)}}(\mathcal{X}_i, \boldsymbol{Z}')\}$
4: Set $\boldsymbol{w}^{(r+1)}$ as the solution of the convex optimization problem (6) (Algorithm 2)
5: **until** convergence to a local minimum
6: **output:** $\boldsymbol{w}^{(r+1)}$

4.6 Solving the relaxed MMS optimization problem using the Pegasos framework

In order to solve the relaxed convex optimization problem (6) efficiently at each round of the CCCP, we have

1. Given a function g, its subgradient $\partial g(\boldsymbol{x})$ at $\boldsymbol{x}$ satisfies: $\forall \boldsymbol{u}, g(\boldsymbol{u}) - g(\boldsymbol{x}) \geq \partial g(\boldsymbol{x}) \cdot (\boldsymbol{u} - \boldsymbol{x})$. The set of all subgradients of g at $\boldsymbol{x}$ is called the subdifferential of g at $\boldsymbol{x}$.

Algorithm 2 Pegasos algorithm for solving the relaxed MMS problem

1: **input:** $\boldsymbol{w}_0$, $\{\mathcal{X}_i, \boldsymbol{Z}_i, \mathcal{C}_i^{(r)}\}_{i=1}^N$, λ, T, K, B
2: **for** $t = 1, 2, \ldots, T$ **do**
3: Draw at random $A_t \subseteq \{1, \ldots, N\}$, with $|A_t| = K$
4: Compute $\hat{\boldsymbol{Z}}_k = \arg\max_{\bar{\boldsymbol{Z}} \notin \mathcal{Z}_k} \big(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}}, \mathcal{Z}_k) + \boldsymbol{w}_t \cdot \Phi(\mathcal{X}_k, \bar{\boldsymbol{Z}})\big) \quad \forall k \in A_t$
5: Set $A_t^+ = \{k \in A_t : \ell_{\text{cccp}}^{(r)}(\mathcal{X}_k, \mathcal{Z}_k; \boldsymbol{w}_t) > 0\}$
6: Set $\boldsymbol{w}_{t+\frac{1}{2}} = (1 - \frac{1}{t})\boldsymbol{w}_t + \frac{1}{\lambda K t} \sum_{k \in A_t^+} \big(\sum_{\boldsymbol{Z} \in \mathcal{C}_i^{(r)}} \Phi(\mathcal{X}_k, \boldsymbol{Z}) / |\mathcal{C}_i^{(r)}| - \Phi(\mathcal{X}_k, \hat{\boldsymbol{Z}}_k)\big)$
7: $\boldsymbol{w}_{t+1} = \min\Big(1, \sqrt{2B/\lambda} / \|\boldsymbol{w}_{t+\frac{1}{2}}\|\Big) \boldsymbol{w}_{t+\frac{1}{2}}$
8: **end for**
9: **output:** $\boldsymbol{w}_{T+1}$

designed a stochastic subgradient descent algorithm, using the Pegasos framework [36]. At each step the algorithm takes K random samples from the training set and calculates an estimate of the subgradient of the objective function using these samples. Then it performs a subgradient descent step with decreasing learning rate, followed by a projection of the solution into the space where the optimal solution lives (line 7). An upper bound on the radius of the ball in which the optimal hyperplane lives can be calculated by considering that

$$\frac{\lambda}{2}\|\boldsymbol{w}^*\|_2^2 \leq \min_{\boldsymbol{w}} \ \frac{\lambda}{2}\|\boldsymbol{w}\|_2^2 + \frac{1}{N}\sum_{i=1}^{N} \ell_{\text{cccp}}^{(r)}\left(\mathcal{X}_i, \mathcal{Z}_i; \boldsymbol{w}\right) \leq B \ ,$$

where $\boldsymbol{w}^*$ is the optimal solution of problem (6), with $B = \max_i(\ell_{\text{cccp}}^{(r)}(X_i, Z_i; \mathbf{0}))$, which equals the maximum number of regions in the image multiplied by the number of attributes P we model. The details of the Pegasos algorithm for solving problem (6) are given in Algorithm 2. Using the theorems in [36] it is easy to show that after $\widetilde{\mathcal{O}}\big(1/(\lambda\varepsilon)\big)$ iterations Algorithm 2 converges in expectation to a solution of accuracy ε.

Efficient implementation: Note that even if we solve the problem in the primal, we can still use nonlinear kernels without computing the nonlinear mapping $\Phi(\cdot, \cdot)$ explicitly. The implementation method is similar to the one described in [36, sec. 4], here we will present it briefly for completeness. In Algorithm 2, $\boldsymbol{w}_{t+\frac{1}{2}}$ can be written as a weighted linear summation of $\Phi(\mathcal{X}_k, \cdot)$. Thus, the algorithm can easily store the coefficient of $\Phi(\mathcal{X}_k, \cdot)$ as well as $\mathcal{X}_k$, $\boldsymbol{Z}$ and $\hat{\boldsymbol{Z}}_k$. In prediction, when we calculate the dot product between $\boldsymbol{w}_t$ and $\Phi(\mathcal{X}_k, \bar{\boldsymbol{Z}})$, we only need to access the dot product between $\Phi(\cdot, \cdot)$. This computation can be further reduced using methods like kernel caching [8].

At each iteration of Algorithm 2, step 4 searches for the most violating labeling vector $\hat{\boldsymbol{Z}}_k$, which is typically computationally expensive. Dynamic programming can be carried out to reduce the computational cost since the contribution of each instance is additive over different labels. In the general situation, the worst case complexity of a naive implementation which enumerates all the possible permutations is $\mathcal{O}(\prod_{m=1}^{M_i} K_{i,m})$, where $K_{i,m}$ is the number of unique possible labels for $\boldsymbol{x}_{i,m}$ in $\mathcal{Z}_i$ (usually $K_{i,m} \ll L_i$). However, the computation time can be further reduced by exploiting the structure of $\mathcal{Z}_i$. This complexity can be greatly reduced when there are special structures such as graphs and trees in the CLSs. See for example [32, sec. 4] for a discussion on some specific problems and special cases. In sec. 6.2, we will present an efficient inference algorithm specialized for solving the name association problem. In cases when computing an exact solution is too expensive, it is often possible to calculate an approximate solution of the same problem, and obtain good empirical results [41] with theoretical guarantees [22].

5 Experiments on Artificial Data

In order to evaluate the proposed algorithm, we first perform experiments on several artificial datasets created from four widely used multi-class datasets taken from the LIBSVM [8] website (usps, letter, news20 and covtype).

The artificial training sets are created as follows: we first set at random pairs of classes as "correlated classes", and as "ambiguous classes", where the ambiguous classes can be different from the correlated classes. Following that, instances are grouped randomly into bags of fixed size B with probability at least P_c that two instances from correlated classes will appear in the same bag. Then L ambiguous labeling vectors are created for each bag, by modifying a few elements of the correct labeling vector. First, the number of elements to modify b is randomly chosen from $\{1, \ldots, B\}$. Then b instances are randomly chosen from the bag, and new labels are randomly chosen among a predefined ambiguous set. The ambiguous set contains the other correct labels from the same bag (except the true one) and a subset of the ambiguous pairs of all the correct labels from the bag. The probability of whether the ambiguous pair of a label is present equals P_a. For testing, we use the original test set, and each instance is considered separately.

Varying P_c, P_a, and L we generate datasets with different difficulty levels to evaluate the behaviour of the algorithms. For example, when $P_a > 0$, noisy labels are likely to be present in the labeling set. Meanwhile, P_c controls the ambiguity within a bag. If P_c is large, instances from two correlated classes are likely to be grouped into the same bag, thus it becomes more difficult to distinguish between them. The parameters P_c and P_a are chosen from $\{0, 0.25, 0.5\}$. For each difficulty level, we use 3 random training/test splits.

For our algorithm, we set the regularization parameter λ to $1/N$ in all of our experiments. We benchmark MMS against the following baselines:

SVM: we train a fully-supervised SVM classifier using the ground-truth labels by considering every instance separately. Its performance is an upper bound of the performance using candidate labeling sets. In all experiments, we use the LIBLINEAR [17] package and test

two different multi-class extensions, the 1-vs-All method with L1-loss (1vA-SVM) and the method by Crammer and Singer [11] (MC-SVM).

CL-SVM: the Candidate Labeling SVM (CL-SVM) is a naive approach which transforms the ambiguously labeled data into a standard supervised representation by treating all possible labels of each instance as true labels. CL-SVM then learns K separate 1-vs-All SVM classifiers from the resulting dataset, where the negative examples for the y-th classifier are instances which do not have the corresponding label y in their candidate labeling set, in other words, it is not possible for these instances to come from class y. A similar baseline has been used in the two-class MIL literature [7].

MIML: we also compared with two SVM-based MIML algorithms[2]: MIMLSVM [45] and M^{3}MIML [44]. We trained the MIML algorithms by treating the labels in $\mathcal{Z}_i$ as a label for the bag. During the test phase, we consider each instance separately and predict the labels as: $y = \arg\max_{y \in \mathcal{Y}} F_{\text{miml}}(\boldsymbol{x}, y)$, where F_{miml} is the classifier learned during training, and $F_{\text{miml}}(\boldsymbol{x}, y)$ can be interpreted as the confidence of the classifier in assigning label y to instance $\boldsymbol{x}$. For a fair comparison, we use the linear kernel in all methods. The cost parameter for SVM algorithms is selected from the range $C \in \{0.1, 1, 10, 100, 1000\}$, and the best results are reported. The bias term is used in all algorithms.

In fig. 3, we plot the average classification accuracy. Several observations can be made. First, MMS achieves results close to the supervised SVM methods, and better than all other baselines. As MMS uses a similar multi-class loss as MC-SVM, it even outperforms 1vA-SVM when the loss has its advantage (e.g. on the 'letter' dataset). For the 'covtype' dataset, the performance gap between MMS and SVM is more visible. It may due to the fact that 'covtype' is a class unbalanced dataset, where the two largest classes (among seven) dominate the whole dataset (more than 85% of the total number of samples). Second, the change in performance of MMS is small when the size of the candidate labeling set grows. Moreover, when correlated instances and extra noisy labels are present in the dataset, the baseline methods' performance drops significantly, whereas MMS is less affected.

The CCCP algorithm usually converges in 3 – 5 rounds, and the final performance is about 5% – 40% higher compared to the results obtained after the first round, especially when L is large. This behavior also proves that approximating the second $\max(\cdot)$ function in the loss function (2) with the average over all the possible labeling vectors can lead to poor performance.

2. We used the original implementation at http://lamda.nju.edu.cn/data.ashx#code. We did not compare against MIMLBOOST [45], because it does not scale to all the experiments we conducted. Besides, MIMLSVM [45] does not scale to data with high dimensional feature vectors (e.g., news20 which has a 62,061-dimensions features). Running the MATLAB implementation of M^{3}MIML [44] on problems with more than a few thousand samples is computational infeasible. Thus, we will only report results using this two baseline methods on small size problems, where they can be finished in a reasonable amount of time.

6 Real case 1: Who is in the picture?

The first real-world problem we tackle is naming faces in news images accompanied by captions written by journalists [4], [26]. Thanks to recent developments in the computer vision and natural language processing fields, generic faces can be localized in the images using a face detector [38] and generic names can be localized in the captions using a named entity detector [1]. Because the names of the most important persons in the image typically appear in the caption, we can attempt to *automatically* label the detected faces with their correct names. This enables to gather a large and realistic face dataset as well as learning face classifiers directly from news items, saving the effort of manually labeling the faces. The main challenge of this task is the *correspondence ambiguity*: there could be multiple faces in the image and/or multiple names in the caption, and not all the names in the caption appear in the image, and vice versa. Some of the face detections can be false positives, which adds to the ambiguity. The task of an algorithm is to resolve the correspondence ambiguity, i.e. assign a name from the caption to each face in the image (or the *null* label if a person is not mentioned in the caption, or for false positive detections).

6.1 Modeling

Since in this problem we are only interested in learning face classifiers, the model will associate at most one label (a name) to each detected face region (no other attributes). In practice, there can be thousands of different names in real world datasets (e.g. the whole of Yahoo! News Dataset [4]). However, users are typically only interested in the top K most frequent names, or only in a limited number of celebrities. Moreover, because of the ambiguities mentioned above, we add a label called *null* (this also covers for those infrequent names we do not model). In total, there are $K+1$ classes, including K face classifiers to be learned.

For each detected face in an image we want to assign a name from the caption to it, or *null*. To format this problem into our framework, we use the following constraints to generate the CLSs:

- [A] a face can be assigned to exactly one name or to *null*;
- [B] a name can be assigned to at most one face;
- [C] a face can be assigned only to a name appearing in the caption of the corresponding image;
- [D] if spatial indicators, such as "left" and "(L)", exist, the labeling vectors in the CLSs should respect them.

In our setup, constraints [A], [B] and [C] apply to all news items, whereas constraint [D] applies only to a few items. When including constraints [A]-[C], the number of admissible candidate labeling vectors for

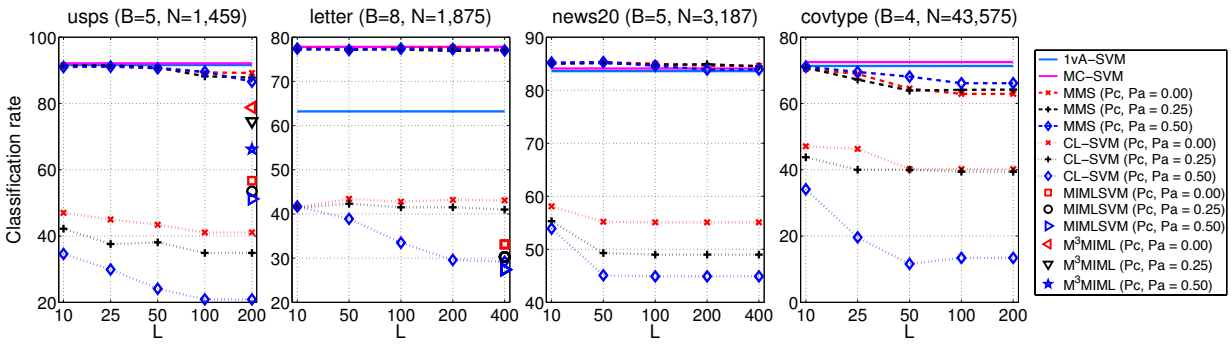


Fig. 3. (**Best seen in color**) Classification performance of different algorithms on artificial datasets.

an image-caption pair with M_i faces and W_i names is $L^i = \sum_{j=0}^{\min(M_i,W_i)} \binom{M_i}{j} \cdot \binom{W^i}{j}$ (see fig. 4 for an example with $M_i = W_i = 2$). In addition, when a spatial indicator is available, we remove the labeling vectors which do not comply with it. Take fig. 2 for example: we decide the relative position of faces by the horizontal coordinate of their center. With the spatial indicators, we know that the face in the middle can only be Grant Hackett. Then the face on the left can either be Igor Chervynsky or Erik Vendt (and the same for the face or the right). Their CLS can be generated as the two persons and two faces case. Different from previous methods, we do not allow the labeling vector which assigns all faces to *null*, because classifying every face as *null* would lead to the trivial solution with 0 loss.

6.2 Inference

As stated above, searching for $\hat{\boldsymbol{Z}}$ in line 4 of Algorithm 2 is the most expensive step of the training procedure. Here we propose an algorithm which can find $\hat{\boldsymbol{Z}}$ efficiently. We first compute the classification scores $s_{\boldsymbol{w}}(x_m, y)$ for every instances in the bag and every possible label $y \in \mathbb{Y}$. After the scores are computed, we use Algorithm 3 to find $\hat{\boldsymbol{Z}}$ in polynomial time with a bounded number of iterations.

The design of Algorithm 3 is motivated by the A* search algorithm [9]. The algorithm first ranks the possible predictions y_m for each face $\boldsymbol{x}_m$ according to their scores $s_{\boldsymbol{w}}(\boldsymbol{x}_m, y_m) + \Delta_A(y, \mathcal{Z}(m))$, where $\mathcal{Z}(m)$ is the m-th row of $\mathcal{Z}$. Lines 9-14 of the algorithm guarantee that all elements in the heap H have a higher score S than any $S(\mathcal{X}, \boldsymbol{Z}) = \sum_{\boldsymbol{x}_m \in \mathcal{X}, z_m \in \boldsymbol{Z}} s_{\boldsymbol{w}}(\boldsymbol{x}_m, z_m)$ for any other arbitrary compositions of $\boldsymbol{Z}$ with $z_m \in \mathbb{Y}$, which not have been added into H. It is easy to verify that the algorithm terminates in at most $L + 1$ iterations, where L is the number of candidate labeling vectors in $\mathcal{Z}$. The worst case scenario is when the first L $\bar{\boldsymbol{Z}}$s returned by the heap (line 8) all belong to $\mathcal{Z}$. As typically $L \gg K$, the worst case complexity of searching for $\hat{\boldsymbol{Z}}$ using Algorithm 3 is $\mathcal{O}(L*M)$, where M is the number of regions in an image. Hence, Algorithm 3 can be used to obtaine $\hat{\boldsymbol{Z}}$ efficiently when the value of L and B is not very large ($L \leq 10^4$ and $M \leq 10$), which is the case in this task.

Algorithm 3 Efficient Algorithm for Searching for $\hat{\boldsymbol{Z}}$

1: **input:** $\mathcal{Z}$, $s(x_m, y) + \Delta_A(y, \mathcal{Z}(m))$, $\forall\ m = 1, \ldots, M,\ y \in \mathbb{Y}$
2: **initialize:** H = new *heap*, index variable $j_m = 1,\ \forall m$
3: Set $\boldsymbol{X}_m$ as a sorted array of y in descending order, according to $s(x_m, y) + \Delta_A(y, \mathcal{Z}_m)$, $\forall y \in \mathbb{Y}$
4: Set $\boldsymbol{Z} = \left[\boldsymbol{X}_1(j_1), \ldots, \boldsymbol{X}_M(j_m)\right]$
5: Set $S = \sum_m s(x_m, \boldsymbol{Z}(m))$
6: Push $\mathcal{A} = \{\boldsymbol{j} = \left[j_1,\ \ldots,\ j_M\right],\ \boldsymbol{Z},\ S\}$ into H
7: **repeat**
8: Pop $\mathcal{A} = \{\boldsymbol{j}, \bar{\boldsymbol{Z}}, S\}$ with the highest score S out of H
9: **for** $m = 1, 2, \ldots, M$ **do**
10: Set $\boldsymbol{j}' = \boldsymbol{j}$, then $\boldsymbol{j}'(m)$++
11: Set $\boldsymbol{Z}' = \left[\boldsymbol{X}_1(\boldsymbol{j}'(1)),\ \ldots,\ \boldsymbol{X}_M(\boldsymbol{j}'(M))\right]$
12: Set $S = \sum_{m'} s(x_{m'}, \boldsymbol{Z}'(m'))\}$
13: Push $\mathcal{A}' = \{\boldsymbol{j}',\ \boldsymbol{Z}',\ S\}$ into H
14: **end for**
15: **until** $\bar{\boldsymbol{Z}} \notin \mathcal{Z}$
16: **output:** $\hat{\boldsymbol{Z}} = \bar{\boldsymbol{Z}}$

In general, Algorithm 3 can not guarantee to find an exact solution $\hat{\boldsymbol{Z}}$ to the problem $\arg\max_{\bar{\boldsymbol{Z}} \notin \mathcal{Z}} U(\bar{\boldsymbol{Z}}) = \arg\max_{\bar{\boldsymbol{Z}} \notin \mathcal{Z}} \left(\boldsymbol{\Delta}_A(\bar{\boldsymbol{Z}}, \mathcal{Z} + \boldsymbol{w}_t \cdot \Phi(\mathcal{X}_k, \bar{\boldsymbol{Z}}))\right)$, denoted by $\hat{\boldsymbol{Z}}^*$, at every round. This is because the algorithm only considers $\boldsymbol{\Delta}_A(y, \mathcal{Z}(m)) = 1$ for those labels that do not appear in the $\mathcal{Z}(m)$. Let us consider a more concrete example: assume a bag $\{\boldsymbol{x}_1, \boldsymbol{x}_2\}$ with two labeling vectors $\boldsymbol{z}_1 = [1, 2]$ and $\boldsymbol{z}_2 = [2, 1]$. We also know that $s(\boldsymbol{x}_1, 1) = 3$, $s(\boldsymbol{x}_1, 2) = 2$, $s(\boldsymbol{x}_1, 3) = 1$, $s(\boldsymbol{x}_2, 1) = s(\boldsymbol{x}_2, 2) = 1$ and $s(\boldsymbol{x}_2, 3) = 0.99$. In this case, the solution obtained by Algorithm 3 is $\hat{\boldsymbol{Z}}$ = [1, 3], but $\hat{\boldsymbol{Z}}^* = [1, 1]$. Despite that, the algorithm will still obtain a $\hat{\boldsymbol{Z}}$ whose value of $U(\hat{\boldsymbol{Z}})$ is very close to $U(\hat{\boldsymbol{Z}}^*)$. However, in practice, the algorithm works very well, and the above special case rarely happens. Experiments on the same dataset show that Algorithm 3 almost always find $\hat{\boldsymbol{Z}}^*$, and that MMS executed with Algorithm 3 achieves the same performance as using an exponential time exact inference algorithm.

6.3 Experiments

We conducted experiments on the Labeled Yahoo! News dataset [4], [26]. The dataset was introduced by Berg et al. [4], and was collected from http://news.yahoo.com/. It consists of news images and their captions describing

the events appearing in the images. Guillaumin et al. [26] provides ground-truth annotations of the dataset and precomputed feature descriptors of the faced detected by [38]. The descriptors are 128-D SIFT [33] at 3 scales at 13 landmark points localized by [16], resulting in a 4992-D descriptor. In our experiments, we only use the first 1664-D features at scale 1.

We compare our results to the same baselines proposed in sec. 5. In MIMLSVM, *null* faces are automatically considered as negative instances. In addition, we also compare with a baseline which does not consider the appearance of the faces:

RANDOM: randomly assign a name (or *null*) from the caption to each face in the corresponding image.

The dataset contains 20071 images and 31147 detected faces. The maximum number of detected faces in an image is 15, and the maximum number of names in a caption is 9. There are more than 10000 different names. We consider two different protocols, detailed in the following sections:

6.3.1 Protocol I.

In the first set of experiments, we use only constraints [A]-[C] to generate the CLS $\mathcal{Z}_i$ of each image-caption pair. We retain the 214 names occurring at least 20 times, and treat the other names as *null*. The experiments are performed over 5 different random train/test splits, sampling 80% of the items as training set and using the rest for testing. During splitting we also maintain the ratio between the number of samples from each class in the training and test set. Performance is measured by how many faces in the test set are correctly labeled with their name (or *null*). Moreover, we also compute name assignment performance on the training set by testing the final model on it.

Table 1 reports the name assignment performance on the training set. All the weakly supervised learning algorithms which consider face appearance outperform the RANDOM baseline, and MMS achieves the best result among all approaches. Table 2 summarizes the generalization performance on the test set. Several observation can be made. First, MMS achieves performance comparable to the fully-supervised SVM algorithms (1vA-SVM, MC-SVM), and it outperforms the other methods which train from ambiguously labeled data (i.e. the captions). MMS even achieves an accuracy 4% higher than 1vA-SVM. This gain may be due to the fact that MMS uses a similar multi-class loss as MC-SVM, whose formulation is advantageous on this dataset. Moreover, we also present the result of MC-SVM trained on only half of the training data (MC-SVM[50%]), while evaluating on the same test set. The result shows that when MMS has more training data, it even outperforms the best fully supervised learning method we consider. This illustrates the promise of our method, as large amounts of image-caption pairs can be easily obtained from the internet, without manual labeling efforts.

6.3.2 Protocol II.

Here we consider all four constraints including the spatial indicators [D] (sec. 6.1). Only 3105 image-caption pairs contain any spatial indicator. We use all of them as part of the training set. In addition, we also randomly sample 6895 image-caption pairs from the dataset, resulting in a training set of 10000 image-caption pairs. All other image-caption pairs form the test set. We retain the 460 names occurring at least 3 times, and treat the other names as *null*.

The results are reported in table 3 and 4. Our MMS algorithm can take advantage of the spatial indicators, and improve performance for both name association on the training set (+3.8%) and face recognition on the test set (+1.4%).

7 Real case 2: Who is doing what?

The second real-world problem we tackle is finding out "who's doing what", i.e. associating names and action verbs in the captions to the faces and body poses of the persons in the images. In addition, the algorithm should also learn visual appearance models for the face and pose classes jointly. This task generalizes the work described in the previous section by considering the subject-verb language construct and by modeling names and verbs jointly. In our previous work [29], we have shown that the correspondence ambiguity is reduced by jointly modeling face and pose together using a generative model. In this section, we show that our MMS technique can be used to model the same problem, and achieves better performance than [29].

7.1 Modeling

In this task, the corpus of news items contains still images of persons performing actions. Each image is annotated with a caption describing "who's doing what" in the image (fig. 1). The interesting regions in the images are persons. A person corresponds to a face and upper-body (including false positive detections), which can be detected with available software. A face and an upper-body are considered to belong to the same person if the face lies near the center of the upper-body bounding-box. One could use a named entity detector [1] and a language parser [13] to extract a list of name-verb pairs from each caption, to represent the connection between a subject and its verb in a sentence. If a name is not connected to any verb, the pair is name-null. Our system models two types of words jointly, and the goals are to: (i) associate the persons in the images to the name-verb pairs in the captions, and (ii) learn a visual appearance model for each name and each verb, corresponding to face and pose classes. These can be used for recognition on new images with or without caption.

The candidate labels for a detected person are the name-verb pairs in the caption. One label assigns a name to a face, and its connected verb from the caption to the

TABLE 1
Protocol I -Overall name assignment accuracy on the training set

Method	RANDOM	CL-SVM	MIMLSVM	MMS
Accuracy	$73.1\% \pm 0.0$	$86.3\% \pm 0.1$	$89.62\% \pm 0.2$	$\mathbf{91.86\% \pm 0.3}$

TABLE 2
Protocol I - Overall face recognition accuracy on the test set

		Supervised Learning			Weakly supervised Learning		
Method	RANDOM	1vA-SVM	MC-SVM	MC-SVM[50%]	CL-SVM	MIMLSVM	MMS
Accuracy	$66.0\% \pm 0.0$	$81.6\% \pm 0.6$	$87.2\% \pm 0.3$	$83.3\% \pm 0.2\%$	$76.9\% \pm 0.2$	$74.7\% \pm 0.9$	$\mathbf{85.7\% \pm 0.5}$

TABLE 3
Protocol II -Overall name assignment accuracy on the training set

	Without POS		With POS	
Method	RANDOM	MMS	RANDOM	MMS
Accuracy	$37.0\% \pm 0.0$	$85.3\% \pm 0.7$	$70.8\% \pm 0.0$	$\mathbf{89.1\% \pm 0.7}$

TABLE 4
Protocol II - Overall face recognition accuracy on the test set

Method	RANDOM	1vA-SVM	MC-SVM	MMS (Without POS)	MMS (With POS)
Accuracy	$39.5\% \pm 0.0$	$82.2\% \pm 0.2$	$87.3\% \pm 0.1$	$83.5\% \pm 0.4$	$\mathbf{84.9\% \pm 0.5}$

$$\mathcal{Z}: \begin{array}{c} \begin{array}{cccccc} \boldsymbol{Z}_1 & \boldsymbol{Z}_2 & \boldsymbol{Z}_3 & \boldsymbol{Z}_4 & \boldsymbol{Z}_5 & \boldsymbol{Z}_6 \end{array} \\ \left[\begin{array}{c|c|c|c|c|c} \mathrm{n}_a & \mathrm{n}_a & \circ & \mathrm{n}_b & \circ & \mathrm{n}_b \\ \mathrm{v}_a & \mathrm{v}_a & \circ & \mathrm{v}_b & \circ & \mathrm{v}_b \\ \mathrm{n}_b & \circ & \mathrm{n}_b & \mathrm{n}_a & \mathrm{n}_a & \circ \\ \mathrm{v}_b & \circ & \mathrm{v}_b & \mathrm{v}_a & \mathrm{v}_a & \circ \end{array}\right] \end{array} \begin{array}{l} \leftarrow \text{person}_1\text{-face} \\ \leftarrow \text{person}_1\text{-pose} \\ \leftarrow \text{person}_2\text{-face} \\ \leftarrow \text{person}_2\text{-pose} \end{array}$$

Fig. 5. CLVs generation for the new item in fig. 1 (left). There are two detected persons, person$_1$ and person$_2$, and two name-verb pairs, Barack Obama-Waving (n_a-v_a) and Michelle Obama-Standing (n_b-v_b). The CLVs are generated using constraints [A]-[C] as in sec. 6.1. Labels such as Barack Obama-Standing is not allowed, as Barack is not the subject of the verb "standing" in the caption.

body pose of the same person in the image. Hence, name and verb are seen as two attributes of the same image region (a person). Therefore, during learning, to find the best possible $\boldsymbol{Z} \in \mathcal{Z}$ for the image, the names and the verbs are considered jointly in making decisions. The chosen name-verb pair is the one which has the highest confidence score over both attributes. For generating constraints, we assume again the uniqueness of each person and her name and apply constraints analog to sec. 6.1 for generating the CLVs (except [D], as spatial indicators are not available in the dataset we perform experiments on). More precisely, the *face* and *name* in constraints [A]-[C] are replaced by *person* and *name-verb*. In this way, each person region is associated with two attributes: name and verb, which must come from the same pair detected from the caption. Fig. 5 illustrates how the CLVs are generated on an example with two persons in the image and two name-verb pairs in the caption (news item in fig. 1 (left)).

7.2 Inference

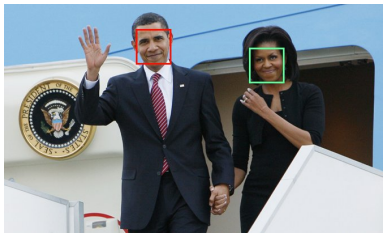
We can use Algorithm 3 to compute $\hat{\boldsymbol{Z}}$.

7.3 Experiments

We conducted experiments on the Idiap/ETHZ Faces and Poses dataset[3]. It contains 1703 image-caption pairs collected by querying Google-images using keywords generated by combining different names (sport stars and politicians) and verbs (from sports and social interactions). An example query is "Barack Obama" + "shake hands". Captions contain the names of some of the persons in the corresponding image, and verbs indicating what they are doing. The captions are derived from the snippet of text returned by Google-images and typically mention the action of at least one person in the image, but also contain names/verbs not appearing in the image. Faces and upper-bodies detected by the methods of [21], [35] were released with the dataset, so we use them directly in our experiments. We also use the available ground-truth name-verb pairs from the captions (instead of running an NLP tool).

For each person region, we extract a face descriptor a body pose descriptor. We describe the face with the method of [16], which detects nine distinctive feature points within the face bounding box. Each point is represented by the pixels in an elliptical region around it, normalized for local photometric invariance. For describing the body poses, we use the features of [20]. A pose E consist of a distribution over the position (spatial and orientation) for each of 6 body parts (head, torso, upper/lower left/right arms) output by the estimator of [15]. Three low-dimensional descriptors are derived from E (e.g. the relative position between pairs of body parts).

3. http://www.vision.ee.ethz.ch/~calvin/faces+poses/



President ***Barack Obama*** *and first lady* ***Michelle Obama*** *wave from the steps of Air Force One as they arrive in Prague, Czech Republic.*

$$\mathcal{Z}: \begin{array}{c|c|c|c|c|c} \boldsymbol{Z}_1 & \boldsymbol{Z}_2 & \boldsymbol{Z}_3 & \boldsymbol{Z}_4 & \boldsymbol{Z}_5 & \boldsymbol{Z}_6 \\ \hline \mathrm{n}_a & \mathrm{n}_a & \circ & \mathrm{n}_b & \circ & \mathrm{n}_b \\ \mathrm{n}_b & \circ & \mathrm{n}_b & \mathrm{n}_a & \mathrm{n}_a & \circ \end{array} \begin{array}{l} \leftarrow \mathrm{face}_1 \\ \leftarrow \mathrm{face}_2 \end{array}$$

Fig. 4. (Left): An example image and its associated caption. There are two detected faces face_1 and face_2 and two names Barack Obama (n_a) and Michelle Obama (n_b) from the caption. (Right): The CLS for this image-captions pairs. The labeling vectors are generated using the constrain [A], [B] and [C], where the *null* class is denoted as $\circ$.

We use non-linear kernels for both face and pose descriptors, in the form $k(\boldsymbol{x}, \boldsymbol{x}') = \exp\left(-\gamma^{-1} d(\boldsymbol{x}, \boldsymbol{x}')\right)$, with d the distance between two descriptors, and γ selected by cross-validation. We measure the distance between two face descriptors $\boldsymbol{x}, \boldsymbol{x}'$ using $d_{\text{face}}(\boldsymbol{x}, \boldsymbol{x}') = 1 - \boldsymbol{x}^T \boldsymbol{x}' / (\|\boldsymbol{x}\| \|\boldsymbol{x}'\|)$. In [20], different similarity measures are proposed for each type of descriptor. We normalize the range of each similarity to $[0, 1]$, and denote their average as $s_{\text{pose}}(\boldsymbol{x}, \boldsymbol{x}')$. The final distance between two poses is $d_{\text{pose}}(\boldsymbol{x}, \boldsymbol{x}') = 1 - s_{\text{pose}}(\boldsymbol{x}, \boldsymbol{x}')$. The face and pose kernel matrices are computed in advance to speed up the learning. It is easy to verify that they are all Mercer kernels.

We compare the results of MMS against (i) a simplified version of the constrained mixture model "GMM" [4, sec. 2.3] which does not incorporate a language model of the caption; (ii) the distance-based generative model "DIST" [29]; (iii) a "RANDOM" baseline which randomly assigns a name-verb pair from the captions to each region in the corresponding image. We did not compare to the other weakly-supervised learning baselines used in sec. 5 & 6 because they do not support multiple attributes. As in the protocol of [29], we use 1600 items for training and 103 for testing.

Fig. 7.3 (left) reports the name and verb assignment performance on the training set, while fig. 7.3 (right) reports the recognition results on the test images. We observe that: (i) DIST using only face information outperforms GMM for name-to-face assignment on the training set. This validates the quality of the distance-based appearance model of [29]. We reuse it also in our MMS framework. (ii) MMS outperforms DIST on both the training and the set set. (iii) the joint "face and pose" model outperforms models using face or pose information alone, demonstrating that modeling both attributes jointly reduces the correspondence ambiguity on the training set and leads to appearance models which perform better on the test set. This phenomenon holds for both DIST and MMS. (iv) on the test set, MMS gets further performance gains when given captions. In this case the problem is easier because the correct label for a person is one of the few appearing in the caption.

8 Conclusions and Future Work

In this paper, we casted the problem of learning from images with captions into a new weakly supervised learning framework. In this framework, each training sample is a bag containing multiple instances, associated with a set of candidate labeling vectors. Each labeling vector encodes the possible labels for the instances in the bag, with only one being fully correct. Our framework provides a principled way to encode different constraints widely used in many tasks, as a list of possible labelings which can be generated automatically from the image-caption pair. We also propose a large margin discriminative learning algorithm to train the classifier effectively. We demonstrate on two real-world tasks that the proposed method can learn face classifiers from images with captions, and can learn multiple attributes jointly (names and verbs).

Our framework can be extended in several ways to solve other tasks which are not demonstrated here. For example, the popular 'image annotation' task: learn visual classifiers for image regions produced by unsupervised segmentation, from images with tags corresponding to nouns [3], [41]. In this task, the size of the candidate labeling set of an image depends exponentially on the number of segments and tags. Therefore, it can be expensive to enumerate all the possible candidate labeling vectors and to infer $\hat{\boldsymbol{Z}}$. In this case, we could use approximate inference to speed up the learning [22]. A recent study on image annotation by Wang and Mori [41] (done independently from ours [30] in the same period) models the problem using the latent SVM framework, and formulates its inference step as a Linear Program (LP) with constraints on tags assignment. It is possible to use a similar LP formulation in our algorithm to perform the inference step approximately without enumerating all the possible assignments.

Acknowledgements: The Labeled Yahoo! News dataset were kindly provided by Matthieu Guillaumin and Jakob Verbeek. We also thank Hugo Penedones for useful discussion on the inference algorithms.

References

[1] http://opennlp.sourceforge.net/.
[2] S. Andrews, I. Tsochantaridis, and T. Hofmann. Support vector machines for multiple-instance learning. In *Proc. NIPS'03*.
[3] K. Barnard, P. Duygulu, D. Forsyth, N. de Freitas, D. Blei, and M. Jordan. Matching words and pictures. *Journal of Machine Learning Research*, 3:1107–1135, 2003.
[4] T. Berg, A. Berg, J. Edwards, and D. Forsyth. Who's in the picture? In *Proc. NIPS'04*.

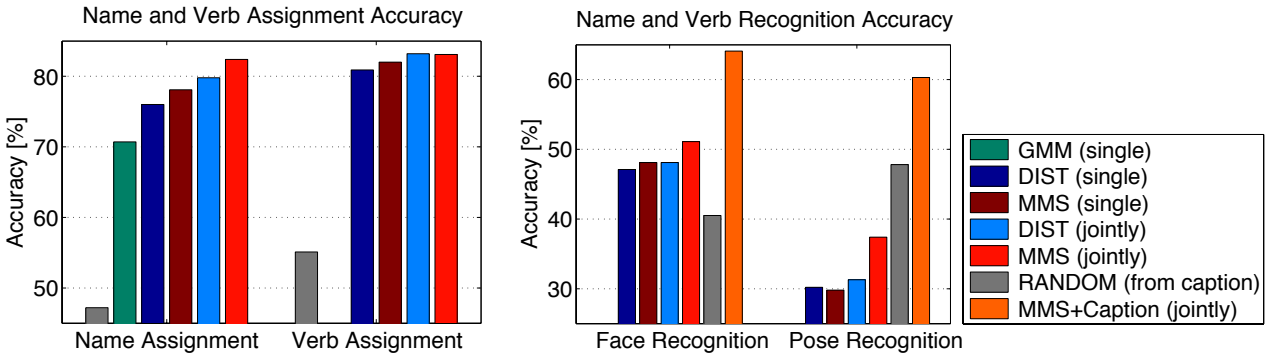


Fig. 6. (**Best seen in color**) (Left) Name and verb assignment accuracy on 1600 training images from Idiap/ETHZ face+pose database. 'Single' stands for modeling one attribute in its corresponding task. 'Jointly' stands for models using two attributes jointly. (Right) Name and verb recognition accuracy on the test images. All methods but 'MMS+Caption' and 'RANDOM (from caption)' only input the image without captions. The 'caption" methods input a test image-caption pair.

[5] D. P. Bertsekas. *Convex Analysis and Optimization*. Athena Scientific, 2003.

[6] M. R. Boutell, J. Luo, X. Shen, and C. M. Brown. Learning multi-label scene classification. *Pattern Recognition*, 37:1537–439, 2004.

[7] R. C. Bunescu and R. J. Mooney. Multiple instance learning for sparse positive bags. In *Proc. ICML'07*.

[8] C. C. Chang and C. J. Lin. *LIBSVM: A Library for Support Vector Machines*, 2001.

[9] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms*. MIT Press, 2003.

[10] T. Cour, B. Sapp, C. Jordan, and B. Taskar. Learning from ambiguously labeled images. In *Proc. CVPR'09*.

[11] K. Crammer and Y. Singer. On the algorithmic implementation of multiclass kernel-based vector machines. *Journal of Machine Learning Research*, 2:265–292, 2001.

[12] N. Cristianini and J. Shawe-Taylor. *An Introduction to Support Vector Machines and other kernel-based learning methods*. Cambridge University Press, 2000.

[13] K. Deschacht and M.-F. Moens. Semi-supervised semantic role labeling using the latent words language model. In *Proc. EMNLP'09*.

[14] T. G. Dietterich, R. H. Lathrop, T. Lozano-Perez, and A. Pharmaceutical. Solving the multiple-instance problem with axis-parallel rectangles. *Artificial Intelligence*, 39:31–71, 1997.

[15] M. Eichner and V. Ferrari. Better appearance models for pictorial structures. In *Proc. BMVC'09*.

[16] M. Everingham, J. Sivic, and A. Zisserman. Hello! my name is... buffy - automatic naming of characters in tv video. In *Proc. BMVC'06*.

[17] R.-E. Fan, K.-W. Chang, C.-J. Lin, S. S. Keerthi, and S. Sundarajan. LIBLINEAR: A library for large linear classification. *Journal of Machine Learning Research*, 9:1871–1874, 2008.

[18] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part based models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 10(9), 2010.

[19] Pedro F. Felzenszwalb and Daniel P. Huttenlocher. Efficient graph-based image segmentation. *International Journal of Computer Vision*, 59(2), 2004.

[20] V. Ferrari, M. Marin, and A. Zisserman. Pose search: retrieving people using their pose. In *Proc. CVPR'09*.

[21] V. Ferrari, M. Marin, and A. Zisserman. Progressive search space reduction for human pose estimation. In *Proc. CVPR'08*.

[22] T. Finley and T. Joachims. Training structural svms when exact inference is intractable. In *Proc. ICML'08*.

[23] Y. Grandvalet. Logistic regression for partial labels. In *Proc. IPMU'02*.

[24] D. Grangier and S. Bengio. A discriminative kernel-based approach to rank images from text queries. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 30(8):1371–1384, 2008.

[25] M. Guillaumin, T. Mensink, J. Verbeek, and C. Schmid. Automatic face naming with caption-based supervision. In *Proc. CVPR'08*.

[26] M. Guillaumin, J. Verbeek, and C. Schmid. Multiple instance metric learning from automatically labeled bags of faces. In *Proc. ECCV'10*.

[27] A. Gupta and L. Davis. Beyond nouns: Exploiting prepositions and comparative adjectives for learning visual classifiers. In *Proc. ECCV'08*.

[28] E. Hüllermeier and J. Beringe. Learning from ambiguously labelled example. *Intelligent Data Analysis*, 10:419–439, 2006.

[29] L. Jie, B. Caputo, and V. Ferrari. Who's doing what: Joint modeling of names and verbs for simultaneous face and pose annotation. In *Proc. NIPS'09*.

[30] L. Jie and F. Orabona. Learning from candidate labeling sets. In *Proc. NIPS'10*.

[31] R. Jin and Z. Ghahramani. Learning with multiple labels. In *Proc. NIPS'02*.

[32] I. Tsochantaridisand T. Joachims, T. Hofmann, and Y. Altun. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[33] D. Lowe. Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60(2):91–110, 2004.

[34] F. Orabona. *DOGMA: a MATLAB toolbox for Online Learning*, 2009. Software available at http://dogma.sourceforge.net.

[35] Y. Rodriguez. *Face Detection and Verification using Local Binary Patterns*. PhD thesis, École Polytechnique Fédérale de Lausanne, 2006.

[36] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal Estimated sub-GrAdient SOlver for SVM. In *Proc. ICML'07*.

[37] A. J. Smola, S. V. N. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *Proc. AISTAT'05*.

[38] P. Viola and M. Jones. Robust real-time object detection. *International Journal of Computer Vision*, 57(2):137154, 2004.

[39] G. Wang and D. Forsyth. Joint learning of visual attributes, object classes, and visual saliency. In *Proc. ICCV'09*.

[40] J. Wang, K. Markert, and M. Everingham. Learning models for object recognition from natural language descriptions. In *BMVC'09*.

[41] Y. Wang and G. Mori. A discriminative latent model of image region and object tag correspondence. In *Proc. NIPS'10*.

[42] C.-N. Yu and T. Joachims. Learning structural svms with latent variables. In *Proc. ICML'09*.

[43] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 15:915–936, 2003.

[44] M.-L. Zhang and Z.-H. Zhou. M^3MIML: A maximum margin method for multi-instance multi-label learning. In *Proc. ICDM'08*.

[45] Z.-H. Zhou and M.-L. Zhang. Multi-instance multi-label learning with application to scene classification. In *Proc. NIPS'06*.

[46] X. Zhu. Semi-supervised learning literature survey. Technical Report 1530, Computer Sciences, University of Wisconsin-Madison, 2005.