Optimizing Average Precision using Weakly Supervised Data

Aseem Behl, C.V. Jawahar, M. Pawan Kumar

Optimizing Average Precision using Weakly Supervised Data

Aseem Behl
IIIT Hyderabad, India

C. V. Jawahar
IIIT Hyderabad, India

M. Pawan Kumar
Ecole Centrale Paris & INRIA Saclay

Abstract

*The performance of binary classification tasks, such as action classification and object detection, is often measured in terms of the average precision (*AP*). Yet it is common practice in computer vision to employ the support vector machine (*SVM*) classifier, which optimizes a surrogate 0-1 loss. The popularity of* SVM *can be attributed to its empirical performance. Specifically, in fully supervised settings,* SVM *tends to provide similar accuracy to the* AP-SVM *classifier, which directly optimizes an* AP*-based loss. However, we hypothesize that in the significantly more challenging and practically useful setting of weakly supervised learning, it becomes crucial to optimize the right accuracy measure. In order to test this hypothesis, we propose a novel latent* AP-SVM *that minimizes a carefully designed upper bound on the* AP*-based loss function over weakly supervised samples. Using publicly available datasets, we demonstrate the advantage of our approach over standard loss-based binary classifiers on two challenging problems: action classification and character recognition.*

1. Introduction

Several problems in computer vision can be formulated as binary classification tasks, that is, determining whether a given input belongs to the positive or the negative class. As a running example throughout this paper, we will consider the task of action classification, that is, automatically figuring out whether an image contains a person performing an action of interest (such as 'jumping' or 'walking'). The importance of binary classification has contributed to the development of several supervised learning approaches, where a binary classifier is estimated using datasets that consist of training samples along with their class information. One such binary classifier that is widely employed in computer vision is the support vector machine (SVM) [22]. Given a fully supervised dataset, an SVM is learned by minimizing a convex regularized upper bound on the 0-1 loss (that is, the loss is 0 for a correct classification and 1 for an incorrect classification).

As the most commonly used accuracy measure for binary classification in computer vision is the average precision (AP) [5], the choice of SVM may appear surprising. Specifically, while AP depends on the ranking of the samples, the 0-1 loss optimized by SVM is only concerned with the number of incorrectly classified samples. The case for its use appears even weaker when we consider that there already exists a related classifier (henceforth referred to as AP-SVM) that optimizes an AP-based loss function (henceforth referred to as the AP loss) [32]. However, a closer look at the empirical evidence reveals the reasoning behind this choice: SVM can be trained more efficiently, and provides comparable accuracy to AP-SVM.

The above observation suggests that we should continue to collect fully supervised datasets and use simple loss functions. If the supervision entails labeling each sample with its class, then this task does not appear to be daunting. However, recent research has shown that the key to achieving high classification accuracy is to provide additional annotations for each sample that can guide the classifier towards the correct output [3, 7, 14, 28, 30]. Going back to the example of action classification, it would be helpful to not only know the class information of each image but the exact location of the person in the image.

The need for complex additional annotations makes supervised learning impractical. To overcome this deficiency, researchers have started exploring weakly supervised learning [1, 4, 7, 10, 11, 15, 18, 17, 19, 24, 25], where the annotations of some or all the samples contain missing information. Not surprisingly, the convenience of using partial annotations comes at the cost of a significantly more challenging machine learning problem. Specifically, weakly supervised learning typically requires us to solve a non-convex optimization problem, which makes it prone to converge to a bad local minimum. Given the inherent difficulty of the problem, we hypothesize that the choice of the loss function becomes crucial in such settings. In order to provide empirical evidence for our hypothesis, we propose a novel latent AP-SVM framework that models the missing additional annotations using latent variables.

Our formulation differs from the standard latent structured SVM (latent SSVM) [31] for general loss functions in three significant aspects. First, it uses a more intuitive two-step prediction criterion, where the first step consists of choosing the best latent variable for each sample and the second step consists of ranking the samples. This is in contrast to the latent SSVM formulation, which requires the joint optimization of the latent variables and the ranking. For example, in 'jumping' action classification, our latent AP-SVM formulation would first pick out the bounding box that is most likely to contain a 'jumping' person

in each image, and then rank them. In contrast, the latent SSVM formulation would require us to simultaneously classify the samples as positive or negative, while picking out the best bounding box for the positive images (bounding box that is most likely to contain a 'jumping' person) and the worst bounding box for the negative images (bounding box that is least likely to contain a 'jumping' person). Second, using the above prediction criterion, the parameters of latent AP-SVM are learned by minimizing a tighter upper bound on the AP loss compared to latent SSVM. Third, unlike latent SSVM, latent AP-SVM lends itself to efficient optimization during learning that is guaranteed to provide a local minimum or saddle point solution. While the first of the aforementioned differences makes our approach more intuitive, the latter two differences provide a sound theoretical justification for its superiority to latent SSVM. In order to demonstrate that the theoretical superiority also translates to better empirical results, we provide a thorough comparison of latent AP-SVM with the baseline methods for two challenging problems: action classification and character recognition. For the sake of clarity, we defer the details that are not essential for the understanding of the paper to the appendices provided in the supplementary material. To facilitate the use of latent AP-SVM, we have made our code and data available online at http://cvit.iiit.ac.in/projects/lapsvm/.

2. Related Work

The popularity of support vector machine (SVM) [22] can be gauged by its numerous applications in computer vision including, image classification [13, 27], action classification [3, 14, 30] and object detection [2, 23]. The main advantages of SVM are its well-understood connections to statistical learning theory [22] and the availability of efficient algorithms to learn its parameters [8, 9, 20].

One of the disadvantages of SVM is that it optimizes the 0-1 loss instead of the average precision (AP) over the training dataset. This disadvantage can be addressed by using the AP-SVM [32] to optimize an upper bound on the AP loss over the training samples. However, empirically, the performance of SVM is comparable to AP-SVM. Furthermore, SVM requires less training time compared to AP-SVM.

Another important disadvantage of SVM is its inability to handle missing information in the annotations. This problem is alleviated by latent SVM [7], which models missing annotations as latent variables. The 0-1 loss based latent SVM can be thought of as a special case of latent structured SVM (latent SSVM) [21, 31], which optimizes a general loss function. Latent SSVM has received considerable attention in the computer vision community [7, 11, 12, 24, 25, 26, 29], on tasks ranging from binary classification (such as object detection) to structured output prediction (such as semantic segmentation and indoor scene understanding). While it can be employed to optimize the AP loss, we will provide both theoretical and empirical arguments for the superiority of our novel latent AP-SVM formulation.

3. Preliminaries

Notation. We use a similar notation to [32]. The training dataset consists of n samples $\mathbf{X} = \{\mathbf{x}_i, i = 1, \cdots, n\}$ together with their class information. The indices for the positive and negative samples are denoted by $\mathcal{P}$ and $\mathcal{N}$ respectively. In other words, if $i \in \mathcal{P}$ and $j \in \mathcal{N}$ then $\mathbf{x}_i$ belongs to the positive class and $\mathbf{x}_j$ belongs to the negative class. Furthermore, for each sample $\mathbf{x}$, the dataset can also provide additional annotations, which we denote by $\mathbf{h}$. For example, in action classification each sample represents an image and the additional annotation $\mathbf{h}$ can represent the bounding box of the person in the image. To simplify the discussion in this section, we will assume that the additional annotations $\mathbf{h}$ are known for all samples. In the next section, we will describe the setting where the additional annotations are latent. We denote the set of all additional annotations for the positive and negative samples by $\mathbf{H}_P = \{\mathbf{h}_i, i \in \mathcal{P}\}$ and $\mathbf{H}_N = \{\mathbf{h}_j, j \in \mathcal{N}\}$ respectively.

The desired output is a ranking matrix $\mathbf{Y}$ of size $n \times n$, such that (i) $\mathbf{Y}_{ij} = 1$ if $\mathbf{x}_i$ is ranked higher than $\mathbf{x}_j$; (ii) $\mathbf{Y}_{ij} = -1$ if $\mathbf{x}_i$ is ranked lower than $\mathbf{x}_j$; and (iii) $\mathbf{Y}_{ij} = 0$ if $\mathbf{x}_i$ and $\mathbf{x}_j$ are assigned the same rank. The ground-truth ranking matrix $\mathbf{Y}^*$ is defined as: (i) $\mathbf{Y}^*_{ij} = 1$ and $\mathbf{Y}^*_{ji} = -1$ for all $i \in \mathcal{P}$ and $j \in \mathcal{N}$; (ii) $\mathbf{Y}^*_{ii'} = 0$ and $\mathbf{Y}^*_{jj'} = 0$ for all $i, i' \in \mathcal{P}$ and $j, j' \in \mathcal{N}$.

AP Loss. Given a training dataset, our aim is to learn a classifier that provides a high AP measure. Let $\mathrm{AP}(\mathbf{Y}, \mathbf{Y}^*)$ denote the AP of the ranking matrix $\mathbf{Y}$ with respect to the true ranking $\mathbf{Y}^*$. The value of the $\mathrm{AP}(\cdot, \cdot)$ lies between 0 and 1, where 0 corresponds to a completely incorrect ranking $-\mathbf{Y}^*$ and 1 corresponds to the correct ranking $\mathbf{Y}^*$. In order to maximize the AP, we will minimize a loss function defined as $\Delta(\mathbf{Y}, \mathbf{Y}^*) = 1 - \mathrm{AP}(\mathbf{Y}, \mathbf{Y}^*)$.

Joint Feature Vector. For positive samples, the feature vector of the input $\mathbf{x}_i$ and additional annotation $\mathbf{h}_i$ is denoted by $\Phi_i(\mathbf{h}_i)$. Similarly, for negative samples, the feature vector of the input $\mathbf{x}_j$ and additional annotation $\mathbf{h}_j$ is denoted by $\Phi_j(\mathbf{h}_j)$. For example, in action classification, $\Phi_i(\mathbf{h}_i)$ can represent poselet [14] or bag-of-visual-words [3] features extracted from an image $\mathbf{x}_i$ using the pixels specified by the bounding box $\mathbf{h}_i$. Similar to [32], we specify a joint feature vector of the input $\mathbf{X}$, output $\mathbf{Y}$, and additional annotations $\{\mathbf{H}_P, \mathbf{H}_N\}$ as

$$\Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} \mathbf{Y}_{ij} (\Phi_i(\mathbf{h}_i) - \Phi_j(\mathbf{h}_j)). \quad (1)$$

In other words, the joint feature vector is the scaled sum of the difference between the features of all pairs of samples, where one sample is positive and the other is negative.

Parameters. The parameter vector of the classifier is denoted by $\mathbf{w}$, and is of the same size as the joint feature vector. Given the parameters $\mathbf{w}$, the ranking of an input $\mathbf{X}$ is defined as the one that maximizes the score, that is,

$$\mathbf{Y}_{opt} = \underset{\mathbf{Y}}{\operatorname{argmax}}\, \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \mathbf{H}), \tag{2}$$

where $\mathbf{H}$ is the set of all the given additional annotations. Yue *et al.* [32] showed that the above optimization can be performed efficiently by sorting the samples $(\mathbf{x}_k, \mathbf{h}_k)$ in descending order of the score $\mathbf{w}^\top \Phi_k(\mathbf{h}_k)$.

Supervised AP-SVM. Given the input $\mathbf{X}$, ranking matrix $\mathbf{Y}$, and additional annotations $\mathbf{H}_P$ and $\mathbf{H}_N$, we would like to learn the parameters of the classifier such that the AP loss over the training dataset is minimized. However, the AP loss is highly non-convex in $\mathbf{w}$, and minimizing it directly can result in a bad local minimum solution. To avoid this undesirable outcome, Yue *et al.* [32] proposed the AP-SVM formulation, which minimizes a regularized upper bound on the AP loss. Specifically, the model parameters are obtained by solving the following convex optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\xi, \\ \text{s.t.} \quad & \forall \mathbf{Y} : \{\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ & -\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\})\} \geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi. \end{aligned} \tag{3}$$

Intuitively, the above problem introduces a margin between the score of the correct ranking and all incorrect rankings. The desired margin is proportional to the difference in their AP values. The hyperparameter C controls the trade-off between the training error and the model complexity.

Problem (3) is specified over all possible rankings $\mathbf{Y}$, which is exponential in the number of training samples. Nonetheless, it can be solved efficiently using a cutting-plane algorithm [32] (described in Appendix A).

4. Optimizing Average Precision with Weak Supervision

The main deficiency of supervised learning is that it involves the onerous task of collecting detailed annotations for each training sample. Since detailed annotations are also very expensive, such an approach quickly becomes financially infeasible as the size of the datasets grow. In this work, we consider a more pragmatic setting where the additional annotations $\mathbf{H}_P$ and $\mathbf{H}_N$ are unknown. For example, consider 'jumping' action classification, where each input represents an image that can belong to the positive class or the negative class. In order to learn a classifier that can distinguish between 'jumping' and 'not jumping' images, we only require image-level annotations instead of the bounding box of the person in each image.

The convenience of not specifying additional annotations comes at the cost of a more complex machine learning problem. Specifically, we need to deal with two confounding factors: (i) since the best value of the additional annotation $\mathbf{h}_i$ for each positive sample $i \in \mathcal{P}$ is unknown, it needs to be imputed automatically; (ii) since a negative sample remains negative regardless of the value of the additional annotation $\mathbf{h}_j$, we need to consider all possible values of $\mathbf{H}_N$ during parameter estimation. In the 'jumping' action classification example, this implies that (i) we have to identify the bounding box of the jumping person in all the positive images, and (ii) ensure that the scores of the identified jumping person bounding boxes are higher than the scores of all possible bounding boxes in the negative images. In the following subsection, we describe how the standard latent SSVM attempts to resolve these confounding factors in order to optimize the AP loss. This will allow us to identify its shortcomings and correct them with our novel formulation in subsection 4.2.

4.1. Latent SSVM Formulation

Given an input $\mathbf{X}$, the prediction rule of a latent SSVM requires us to maximize the score jointly over the output $\mathbf{Y}$ and the additional annotations $\mathbf{H}$, that is,

$$(\mathbf{Y}_{opt}, \mathbf{H}_{opt}) = \underset{(\mathbf{Y},\mathbf{H})}{\operatorname{argmax}}\, \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \mathbf{H}). \tag{4}$$

The parameters $\mathbf{w}$ of a latent SSVM are learned by minimizing a regularized upper bound on the training loss. Specifically, the parameters are obtained by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\xi, \\ \text{s.t.} \quad & \forall \mathbf{Y}, \mathbf{H} : \max_{\hat{\mathbf{H}}} \{\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \hat{\mathbf{H}})\} \\ & -\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \mathbf{H}) \geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi. \end{aligned} \tag{5}$$

Intuitively, the above problem introduces a margin between the maximum score corresponding to the ground-truth output and all other pairs of output and additional annotations. Similar to the supervised setting, the desired margin is proportional to the AP loss.

There are three main drawbacks of the standard latent SSVM formulation in the case of AP loss optimization. The first drawback is the prediction rule. This is specified by problem (4), which requires us to simultaneously label the samples as positive or negative (optimize over $\mathbf{Y}$) and find the highest scoring additional annotations for the positive samples and the lowest scoring additional annotations for the negative samples (optimize over $\mathbf{H}$) in order to maximize the score. This is in stark contrast to the prediction rule of existing weakly supervised binary classifiers, which first obtain the score of each sample by maximizing over the additional annotations (regardless of whether they will be labeled as positive or negative), and then ranking them

according to their scores. For example, in action classification, we rank the images according to the highest scoring bounding box of a person in each image. In other words, we never compare the scores of particular choice of additional annotations with a different set of additional annotations. The second drawback is the learning formulation. This is specified by problem (5), which provides a very loose upper bound on the AP loss. The third drawback is the optimization. Specifically, to the best of our knowledge, the local optimum solution of problem (5) cannot be found efficiently due to the lack of an appropriate cutting plane algorithm. For the details on the difficulty of optimization of latent SSVM, as well as an approximate algorithm used in our experiments, we refer the reader to Appendix C.

4.2. Latent AP-SVM Formulation

We now describe a novel latent AP-SVM formulation that overcomes the three drawbacks of the standard latent SSVM framework discussed in the previous section. Specifically, latent AP-SVM uses an intuitive prediction rule, provides a tighter upper bound on the AP loss, and lends itself to efficient optimization.

4.2.1 Intuitive Prediction

We use a two-step prediction rule. In the first step, we obtain the value of the additional annotations for each sample by maximizing the score, that is,

$$\mathbf{h}_{opt} = \underset{\mathbf{h}}{\operatorname{argmax}}\, \mathbf{w}^\top \Phi(\mathbf{h}). \tag{6}$$

Next, we obtain the optimal ranking $\mathbf{Y}_{opt}$ for the additional annotations $\mathbf{H}_{opt}$, that is,

$$\mathbf{Y}_{opt} = \underset{\mathbf{Y}}{\operatorname{argmax}}\, \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \mathbf{H}_{opt}), \tag{7}$$

where $\mathbf{H}_{opt}$ is the set of all the additional annotations obtained by solving problem (6) for all samples. Similar to the supervised setting, the optimal ranking is computed by sorting the samples in descending order of their scores. Note that our prediction rule is the same as the ones used in conjunction with the current weakly supervised binary classifiers [1, 7].

4.2.2 Tighter Bound on the AP Loss

We learn the parameters of latent AP-SVM by solving the following optimization problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2}||\mathbf{w}||^2 + C\xi, \\ \text{s.t.} \quad & \forall \mathbf{Y}, \mathbf{H}_N : \max_{\mathbf{H}_P} \{\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ & -\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\})\} \geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi. \end{aligned} \tag{8}$$

Intuitively, the above problem finds the best assignment of values for the additional annotations $\mathbf{H}_P$ of the positive samples such that the score for the correct ranking (which places all the positive samples above the negative samples) is higher than the score for an incorrect ranking, regardless of the choice of the additional annotations $\mathbf{H}_N$ of the negative samples.

It is worth noting the significant difference between the optimization corresponding to latent AP-SVM and the standard latent SSVM. Specifically, in the constraints of problem (5), the values of the additional annotations for a correct and incorrect ranking are independent of each other. In contrast, the constraints of problem (8) are specified using the same values of the additional annotations. The following proposition provides a sound theoretical justification for preferring problem (8) over problem (5).

Proposition 1. *The latent* AP-SVM *formulation provides a tighter upper bound on the* AP *loss compared to the standard latent* SSVM *formulation (proof in Appendix D).*

4.2.3 Efficient Optimization

The local minimum or saddle point solution of problem (8) can be obtained using the CCCP algorithm [33], as described in Algorithm 1. The algorithm involves two main steps. In the first step (step 3 of Algorithm 1), it imputes the best additional annotations $\mathbf{H}_P$ of the positive samples given the current estimate of the parameters. In the second step (step 4 of Algorithm 1), given the imputed values of $\mathbf{H}_P$, CCCP updates the parameters by solving the resulting convex optimization problem. We discuss both these steps in detail below.

Algorithm 1 *The* CCCP *algorithm for parameter estimation of latent* AP-SVM.

Require: $\mathbf{X}, \mathbf{Y}^*, \mathbf{w}_0, \epsilon$

1: $t \Leftarrow 0$
2: **repeat**
3: For the current set of parameters $\mathbf{w}_t$, obtain the value of the latent variables $\mathbf{H}_P^*$ that minimizes the objective function value of problem (8).
4: Update $\mathbf{w}_{t+1}$ by fixing the latent variables to $\mathbf{H}_P^*$ and solving the resulting convex problem.
5: $t \leftarrow t + 1$
6: **until** Objective function cannot be decreased below tolerance ϵ

Imputing the Additional Annotations. For a given parameter $\mathbf{w}$, we need to obtain the values of the additional annotations $\mathbf{H}_P$ for the positive samples such that it minimizes the objective function of problem (8). Since $\mathbf{w}$ is fixed, the first term of the objective function (that is, the squared ℓ_2 norm of $\mathbf{w}$) cannot be modified. Instead, we need to minimize the slack ξ, which is equivalent to solving the following problem:

$$\begin{aligned} \min_{\mathbf{H}_P} \max_{\mathbf{Y}, \mathbf{H}_N} \{\Delta(\mathbf{Y}^*, \mathbf{Y}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ + \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\})\}. \end{aligned} \tag{9}$$

We refer to the above problem as output-consistent inference (since it fills in the missing information under the constraint that it is consistent with the output, that is, the optimal ranking). Although problem (9) contains $\mathbf{Y}$ and $\mathbf{H}_N$, the following proposition shows that it can be optimized easily with respect to $\mathbf{H}_P$.

Proposition 2. *Problem (9) can be solved efficiently by independently choosing the latent variable for each positive sample using the following criterion:*

$$\mathbf{h}_i^* = \underset{\mathbf{h}_i}{\operatorname{argmax}}\, \mathbf{w}^\top \Phi_i(\mathbf{h}_i), \forall i \in \mathcal{P} \tag{10}$$

(proof in Appendix E).

Updating the Parameters. Given the imputed latent variables $\mathbf{H}_P^*$, the parameters are updated by solving the following convex problem:

$$\begin{aligned} &\min_{\mathbf{w}} \frac{1}{2}||\mathbf{w}||^2 + C\xi, \\ &\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P^*, \mathbf{H}_N\}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P^*, \mathbf{H}_N\}) \\ &\geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi, \forall \mathbf{Y}, \mathbf{H}_N. \end{aligned} \tag{11}$$

Similar to supervised AP-SVM, the above problem can be solved using a cutting plane algorithm. The computational feasibility of the cutting plane algorithm relies on being able to efficiently compute the most violated constraint. In our case, the most violated constraint is found by solving the following problem:

$$\begin{aligned} \hat{\mathbf{Y}}, \hat{\mathbf{H}}_N = \underset{\mathbf{Y}, \mathbf{H}_N}{\operatorname{argmax}} \{&\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P^*, \mathbf{H}_N\}) \\ &- \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P^*, \mathbf{H}_N\}) + \Delta(\mathbf{Y}^*, \mathbf{Y})\}. \end{aligned} \tag{12}$$

We refer to the above problem as loss-augmented inference (since it augments the score of the ranking with its AP loss). Note that, in contrast to supervised AP-SVM, we not only need to optimize over the ranking $\mathbf{Y}$, but also the variables $\mathbf{H}_N$. The following proposition allows us to perform the joint optimization efficiently.

Proposition 3. *Problem (12) can be solved by first maximizing over $\mathbf{H}_N$ using the following criterion:*

$$\mathbf{h}_j^* = \underset{\mathbf{h}_j}{\operatorname{argmax}}\, \mathbf{w}^\top \Phi_j(\mathbf{h}_j), \forall j \in \mathcal{N} \tag{13}$$

(proof described in Appendix E).

Using Proposition 3, problem (12) can be solved in two steps. In the first step we maximize the loss-augmented score over $\mathbf{H}_N$ by maximizing the score of each negative sample independently. The second step is to maximize the loss-augmented score over $\mathbf{Y}$, which is achieved using the optimal greedy algorithm of Yue *et al.* [32] (described in Appendix B).

5. Experiments

The previous section shows the theoretical benefit of latent AP-SVM over the standard latent SSVM formulation, namely that it minimizes a tighter upper bound on the AP loss and allows for efficient inference, while using an intuitive prediction rule. We now show that the theoretical benefits translate to improved empirical performance using two important and challenging problems in computer vision.

5.1. Action Classification

Dataset. We use the PASCAL VOC 2011 [5] action classification dataset, which consists of 4846 images depicting 10 action classes. The dataset is divided into two subsets: 2424 'trainval' images for which we are provided the bounding boxes of the persons in the image together with their action class; and 2422 'test' images for which we are only provided with the person bounding boxes.

Recall that our main hypothesis is that the challenging nature of weakly supervised learning makes it essential to use the right loss function during training. In order to test this hypothesis, we use the 'trainval' images to create five types of datasets that vary in their level of supervision. Specifically, each type of dataset provides the ground-truth additional annotations[1] for S percent of the positive and the negative samples, where $S \in \{0, 25, 50, 75, 100\}$. The additional annotations for the remaining $100-S$ percent of the samples are treated as latent variables. The putative values of each latent variable are restricted to the top $T = 20$ boxes obtained by a standard person detector [6]. During testing, we use the learned parameters to classify the given person bounding boxes in the 'test' dataset. The performance is measured by submitting the scores of all the bounding boxes to the PASCAL VOC evaluation server.

Features. Given a bounding box $\mathbf{h}_i$ of the image $\mathbf{x}_i$, we use the standard poselet-based feature vector [14] to specify $\Phi_i(\mathbf{h}_i)$. It consists of 2400 activation scores of action-specific poselets and 4 object activation scores. In addition, we use the score of the person detector [6], which results in a 2405 dimensional feature vector.

Methods. We compare our latent AP-SVM formulation with the baseline latent SVM that is commonly used in computer vision. Latent SVM consists of two hyperparameters: (i) C, the trade-off between the regularization and the loss; and (ii) J, the relative weight of the positive samples. In order to further strengthen the baseline, we add robustness to outliers using a further hyperparameter c. Specifically, we prevent the classifier from considering the most confusing $c\%$ bounding boxes in the negative samples under the constraint that at least one bounding box is used per negative image. We obtain the best

[1] Additional annotation provided is the bounding-box obtained by a standard person detector overlapping most with the ground-truth bounding box in PASCAL VOC.

settings of the hyperparameters via a 5-fold cross validation, where the 'trainval' set is split into 1940 training images and 484 validation images. We consider the following putative values: $C \in \{10^{-3}, 10^{-2}, \ldots, 10^{4}\}$, $J \in \frac{|\mathcal{P}|+|\mathcal{N}|}{|\mathcal{P}|} \times \{10^{-4}, 10^{-3}, \ldots, 10^{1}\}$ and $c \in \{0, 0.1, \ldots, 0.9\}$ (note that, when $c = 0$, the resulting baseline is the standard latent SVM without robustness). In addition, we also compare the performance of our latent AP-SVM with latent SSVM. For the latent AP-SVM and latent SSVM, we only need to specify a single hyperparameter C, whose value is also obtained via 5-fold cross-validation. In order to mitigate the effects of initialization, we use 5 random seeds and choose the one that provides the minimum objective value for each method independently.

Complexity. The running time for latent AP-SVM and the baseline methods is dominated by computation of the most violated constraint. Empirically, we found that computation of most violated constraint in latent AP-SVM is around 5 times slower and 100 times faster compared to latent SVM and latent SSVM respectively. However, latent AP-SVM does not require any extra hyperparameter to weight the positive samples, therefore does not worsen the overall computational complexity compared to training latent SVM.

Results. Figure 1 shows the best mean AP value over all 10 action classes obtained during 5-fold cross validation. Note that as the amount of supervision decreases, the gap between our method and the two baselines steadily increases. In the fully supervised setting, that is, $S = 100$, latent AP-SVM provides statistically significant improvements over latent SVM for only 4 out of 10 classes (using paired t-test with p-value less than 0.05), with an overall improvement of less than 3%. Note that, for fully supervised datasets, both latent AP-SVM and latent SSVM are equivalent to the AP-SVM, and hence provide the same results. However, in the more interesting weakly supervised setting, that is, $S = 0$, latent AP-SVM provides statistically significant improvements over latent SVM for 6 out of 10 classes, and an overall improvement of more than 5%. We note that, for standard latent SVM, i.e., when c = 0, the mean AP obtained during cross-validation over all ten classes is 39.5%. Adding robustness and cross-validating c, we get an overall improvement of 2.4%. This illustrates that, by incorporating robustness, we get a stronger baseline than standard latent SVM. Similarly, latent AP-SVM provides statistically significant improvements over latent SSVM for 7 out of 10 classes and an overall improvement of more than 4%.

Table 1 shows the comparison of our latent AP-SVM with latent SVM and latent SSVM on the test set. Note that we use 5 different random seeds for each method. The hyperparameters are set using 5-fold cross-validation. Latent AP-SVM classifier performs better than latent SVM for all 10 classes with significant increase in performance for 4 classes. Overall, we get an improvement of 5.1% on the test performance compared to latent SVM. Similarly, latent AP-SVM classifier performs better than latent SSVM for 8 out of 10 classes. Overall, we get an improvement of 3.7% on the test performance compared to latent SSVM.

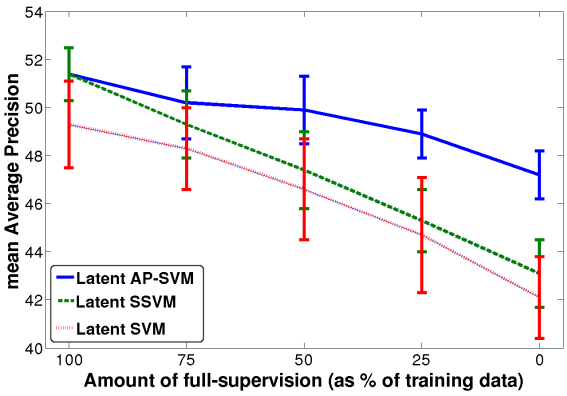

Figure 1. *The best mean average precision over all 10 action classes obtained during 5-fold cross validation. The x-axis corresponds to the amount of supervision provided. The y axis corresponds to the mean average precision. As the amount of supervision decreases, the gap in the performance of latent* AP-SVM *and the baseline methods increases, thereby illustrating the importance of using the correct loss function and the correct learning formulation for weakly supervised learning.*

Figure 2 exemplifies the significance of using the right accuracy measure. It shows a sub-sequence of images from two rankings predicted for the 'ridingbike' action. The top and bottom rows are from the rankings predicted by latent SVM and latent AP-SVM respectively. Latent SVM predicts 2 incorrect images out of 4 but makes mistakes in the top two images. Whereas, latent AP-SVM makes the same number of mistakes but for images at lower rankings (first and fourth in this example). This may be explained by the fact that, in terms of 0/1 loss both predictions are equivalent. However, the bottom ranking sequence is preferred by a classifier that is attempting to optimize the AP loss.

Figure 2. *Top and bottom rows show a sub-sequence of images from rankings predicted by latent* SVM *and latent* AP-SVM *respectively for 'ridingbike' action class. Both first and second row have two incorrect (highlighted by red border) and correct (highlighted by green border) images each, but latent* AP-SVM *puts the incorrect images in the lower rank.*

Method	Jump	Use phone	Play instrument	Read	Ride bike	Ride horse	Run	Take photo	Use computer	Walk	Overall
Latent AP-SVM	45.7	30.5	34.0	21.1	75.5	74.9	76.0	15.7	24.6	47.5	44.6
Latent SSVM	37.6	26.5	33.9	22.5	71.2	66.7	66.8	17.4	21.9	44.8	40.9
Latent SVM	36.9	28.0	32.2	20.6	65.3	68.2	63.5	13.4	21.6	45.7	39.5

Table 1. *The average precision of latent* AP-SVM *and the baseline latent* SVM *and latent* SSVM *methods under weak supervision. The training is performed over the entire 'trainval' dataset with $S = 0$ using the best hyperparameters obtained during 5-fold cross-validation. The testing is performed on the 'test' dataset and evaluated on the* PASCAL VOC *server. The last column ('Overall') shows the mean average precision over all ten action classes.*

5.2. Character Recognition in Natural Images

Dataset. We use the IIIT 5K-WORD [16] scene text dataset, which consists of 5000 cropped word images from scene texts and born-digital images, which are divided into 2000 'trainval' images and 3000 'test' images. Each image is annotated with the corresponding word, that is, a string where each character is an upper case letter ('A' to 'Z'), a lower case letter ('a' to 'z'), or a number ('0' to '9'). In addition, the dataset also provides the bounding boxes for each character of the word, which we discard during learning. Instead, we treat the bounding box of the characters as latent variables whose putative values are restricted to $T = 20$ boxes obtained by a standard character detector [2]. Using this dataset, we perform binary classification for the 22 classes that contain at least 150 samples in the 'trainval' dataset.

Features. Given a character bounding box $\mathbf{h}_i$ of the word image $\mathbf{x}_i$, we use the histogram of oriented gradients (HOG) [2] features to specify $\Phi_i(\mathbf{h}_i)$. The HOG features are computed by resizing the bounding box to 48×48 pixels.

Methods. We compare our latent AP-SVM formulation with the baseline latent SVM. We use the same formulation of the baseline latent SVM as described in the action classification experiments. All the hyper-parameters are once again fixed using 5-fold cross validation over 80%/20% splits of the 'trainval' dataset. Each method is initialized using 3 different random seeds and the solution corresponding to the minimum objective value is chosen.

Results. Figure 3 shows the best AP values for all the 5 classes where the performance of latent AP-SVM is statistically different from that of latent SVM (using paired t-test with p-value less than 0.05). Latent SVM provides statistically significant improvements over latent AP-SVM for only 1 class. In contrast, latent AP-SVM improves the performance for 4 classes. In terms of the mean AP value, latent AP-SVM provides an improvement of 6.3% for the 5 classes shown in figure 3 and 3.2% over all 22 classes.

Figure 4 shows the AP values for the 5 statistically significant characters on the 'test' set. Similar to the cross-validation results, latent AP-SVM outperforms latent SVM for 4 out of the 5 classes. In terms of the mean AP value, latent AP-SVM provides an improvement of 3.5% for the 5 classes shown in figure 4 and 2.7% over all 22 classes. Comparison with latent SSVM and detailed results over all 22 classes are provided in the supplementary material.

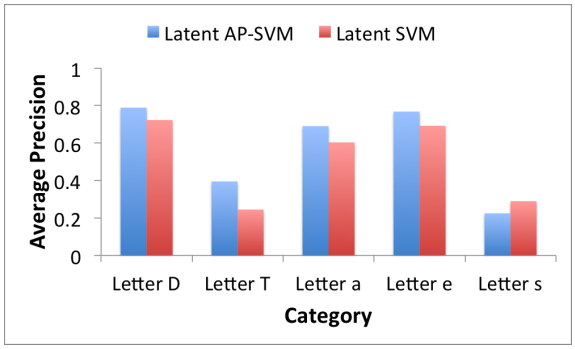

Figure 3. *The best average precision values for the 5 statistically significant character classes obtained during 5-fold cross validation on the 'trainval' set of the* IIIT 5K-WORD *dataset. The x-axis corresponds to the characters. The y axis corresponds to the average precision. Latent* AP-SVM *provides statistically significant improvements over latent* SVM *for 4 out of the 5 characters.*

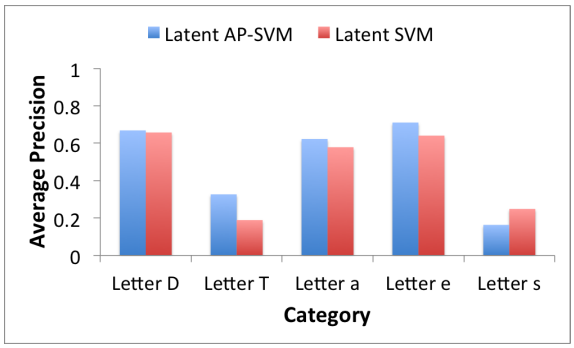

Figure 4. *The average precision values for the 5 statistically significant characters obtained on the 'test' set of the* IIIT 5K-WORD *dataset. The x-axis corresponds to the character categories. The y axis corresponds to the average precision.*

6. Discussion

We proposed a novel latent AP-SVM formulation that allows us to learn accurate classifier parameters by minimizing a carefully designed difference-of-convex upper bound

on the AP loss. We showed the advantage of our approach over latent SVM and the standard latent SSVM for action classification and character recognition using standard, publicly available datasets.

An interesting direction of future research would be to extend the latent AP-SVM formulation to learn from images that have been labeled by noisy tags. This will allow us to exploit the large, freely available datasets provided by photo-sharing websites (for example, Flickr or Picasa). The large size of such datasets would also make it necessary to improve the efficiency of the CCCP.

7. Acknowledgements

This work is partially funded by the European Research Council under the European Community's Seventh Framework Programme (FP7/2007-2013)/ERC Grant agreement number 259112, and the INRIA International Internship Programme.

References

[1] M. Blaschko, A. Vedaldi, and A. Zisserman. Simultaneous object detection and ranking with weak supervision. In *NIPS*, 2010. 1, 4

[2] N. Dalal and B. Triggs. Histograms of oriented gradients for human detection. In *CVPR*, 2005. 2, 7

[3] V. Delaitre, I. Laptev, and J. Sivic. Recognizing human actions in still images: a study of bag-of-features and part-based representations. In *BMVC*, 2010. 1, 2

[4] T. Deselaers, B. Alexe, and V. Ferrari. Localizing objects while learning their appearance. In *ECCV*, 2010. 1

[5] M. Everingham, L. Van Gool, C. Williams, J. Winn, and A. Zisserman. The pascal visual object classes (voc) challenge. *IJCV*, 2010. 1, 5

[6] P. Felzenszwalb, R. Girshick, and D. McAllester. Discriminatively trained deformable part models, release 4. http://people.cs.uchicago.edu/ pff/latent-release4/. 5

[7] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan. Object detection with discriminatively trained part-based models. *PAMI*, 2010. 1, 2, 4

[8] T. Joachims. Making large-scale SVM learning practical. In B. Scholkopf, C. Burges, and A. Smola, editors, *Advances in Kernel Methods*. MIT Press, 1999. 2

[9] T. Joachims, T. Finley, and C.-N. J. Yu. Cutting-plane training of structural svms. *Machine Learning*, 2009. 2

[10] M. P. Kumar, B. Packer, and D. Koller. Modeling latent variable uncertainty for loss-based learning. In *ICML*, 2012. 1

[11] M. P. Kumar, H. Turki, D. Preston, and D. Koller. Learning specific-class segmentation from diverse data. In *ICCV*, 2011. 1, 2

[12] T. Lan, Y. Wang, W. Yang, S. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *PAMI*, 2012. 2

[13] Y. Lin, F. Lv, S. Zhu, M. Yang, T. Cour, K. Yu, L. Cao, and T. Huang. Large-scale image classification: Fast feature extraction and svm training. In *CVPR*, 2011. 2

[14] S. Maji, L. Bourdev, and J. Malik. Action recognition from a distributed representation of pose and appearance. In *CVPR*, 2011. 1, 2, 5

[15] K. Miller, M. P. Kumar, B. Packer, D. Goodman, and D. Koller. Max-margin min-entropy models. In *AISTATS*, 2012. 1

[16] A. Mishra, K. Alahari, and C. V. Jawahar. Scene text recognition using higher order language priors. In *BMVC*, 2012. 6

[17] M. Pandey and S. Lazebnik. Scene recognition and weakly supervised object localization with deformable part-based models. In *ICCV*, 2011. 1

[18] A. Prest, C. Schmid, and V. Ferrari. Weakly supervised learning of interactions between humans and objects. *PAMI*, 2012. 1

[19] O. Russakovsky, Y. Lin, K. Yu, and L. Fei-Fei. Object-centric spatial pooling for image classification. In *ECCV*, 2012. 1

[20] S. Shalev-Shwartz, Y. Singer, and N. Srebro. Pegasos: Primal estimated sub-gradient solver for SVM. In *ICML*, 2009. 2

[21] A. Smola, S. Vishwanathan, and T. Hofmann. Kernel methods for missing variables. In *AISTATS*, 2005. 2

[22] V. Vapnik. *Statistical learning theory*. Wiley, 1998. 1, 2

[23] A. Vedaldi, V. Gulshan, M. Varma, and A. Zisserman. Multiple kernels for object detection. In *ICCV*, 2009. 2

[24] A. Vezhnevets, J. Buhmann, and V. Ferrari. Weakly supervised structured output learning for semantic segmentation. In *CVPR*, 2012. 1, 2

[25] H. Wang, S. Gould, and D. Koller. Discriminative learning with latent variables for cluttered indoor scene understanding. In *ECCV*, 2010. 1, 2

[26] Y. Wang and G. Mori. A discriminative latent model of object classes and attributes. In *ECCV*, 2010. 2

[27] J. Yang, K. Yu, Y. Gong, and T. Huang. Linear spatial pyramid matching using sparse coding for image classification. In *CVPR*, 2009. 2

[28] L. Yang, R. Jin, R. Sukthankar, and F. Jurie. Unifying discriminative visual codebook generation with classifier training for object category recognition. In *CVPR*, 2008. 1

[29] W. Yang, Y. Wang, and G. Mori. Recognizing human actions from still images with latent poses. In *CVPR*, 2010. 2

[30] B. Yao, X. Jiang, A. Khosla, A. Lin, L. Guibas, and L. Fei-Fei. Human action recognition by learning bases of action attributes and parts. In *ICCV*, 2011. 1, 2

[31] C.-N. Yu and T. Joachims. Learning structural svms with latent variables. In *ICML*, 2009. 1, 2

[32] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support vector method for optimizing average precision. In *SIGIR*, 2007. 1, 2, 3, 5

[33] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 2003. 4