

Supplementary Material - 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

1. Appendix

A. Optimization for AP-SVM

Algorithm 1 *Cutting plane algorithm for solving AP-SVM.*

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

- 1: Initialize the set of active constraints $\mathcal{W} \leftarrow \emptyset$
 - 2: $t \leftarrow 0$
 - 3: **repeat**
 - 4: $t \leftarrow t + 1$
 - 5: Learn parameters $\mathbf{w}_t, \xi_t$ by solving the following convex problem over the set of active constraints $\mathcal{W}$,

$$\underset{\mathbf{w}, \xi}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{w}\|^2 + C\xi$$

$$\text{s.t. } \forall \mathbf{Y} \in \mathcal{W} : \{\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\})\} \geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi.$$
 - 6: Find the most violated constraint by solving the following problem,

$$\hat{\mathbf{Y}} = \underset{\mathbf{Y}}{\operatorname{argmax}} \{\Delta(\mathbf{Y}^*, \mathbf{Y}) + \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\})\} \quad (1)$$
 - 7: Add the most violated constraint to set of active constraints $\mathcal{W}$.
 - 8: **until**

$$\Delta(\mathbf{Y}^*, \hat{\mathbf{Y}}) + \{\mathbf{w}_t^\top \Psi(\mathbf{X}, \hat{\mathbf{Y}}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \mathbf{w}_t^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\})\} \leq \xi_t + \epsilon.$$
-

Algorithm 1 describes a cutting plane algorithm for estimating the parameters of an AP-SVM by solving the following problem:

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

Briefly, the algorithm starts by specifying no constraints (step 1 of Algorithm 1: $\mathcal{W}$ is initialized to the null set). At each iteration, it adds a single constraint, which corresponds to the most violated ranking (step 4 of Algorithm 1: solving problem (1)). Intuitively, problem (1) finds a ranking that differs significantly from the ground-truth ranking

in terms of AP but has a high score. Having added the most violated constraint, the cutting plane algorithm updates the parameters by solving a convex quadratic program (step 3 of Algorithm 1). The algorithm stops once no constraint can be found that is violated by more than the desired precision ϵ .

The feasibility of the cutting plane algorithm relies on solving problem (1) efficiently. For fixed values of $\mathbf{H}_P$ and $\mathbf{H}_N$ (which is indeed the case for supervised AP-SVM), this can be achieved using the algorithm described in the following appendix.

B. Loss-Augmented Inference for Fixed Additional Annotations

Algorithm 2 *Finding the most violated constraint for AP loss with known values of additional annotations.*

Require: $\mathbf{X}, \mathbf{Y}^*, \mathbf{H}_P, \mathbf{H}_N, \mathbf{w}$

- 1: Sort positive inputs $\mathbf{x}_i \in \mathcal{P}$ and negative inputs $\mathbf{x}_j \in \mathcal{N}$ in descending order of classifier score $\mathbf{w}^\top \Phi_i(\mathbf{h}_i)$ and $\mathbf{w}^\top \Phi_j(\mathbf{h}_j)$ respectively.
 - 2: Initialize ranking $\mathbf{Y}$, s.t. all positive samples are ranked higher than negative samples.
 - 3: **for** $j = 1 \rightarrow |\mathcal{N}|$ **do**
 - 4: Find the position of j^{th} ranked negative which maximizes loss-augmented score,

$$\Delta(\mathbf{Y}^*, \mathbf{Y}) + \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}).$$
 - 5: Update $\mathbf{Y}$ with the best position of the j^{th} ranked negative.
 - 6: **end for**
 - 7: **return** $\mathbf{Y}$.
-

The greedy algorithm of Yue *et al.* [2] for solving problem (1) for a fixed values of $\mathbf{H}_P$ and $\mathbf{H}_N$ is outlined in Algorithm 2. Briefly, it starts with the ground-truth ranking, where each positive sample is ranked higher than all the negative samples (step 2 of Algorithm 2). Next, it finds the ranking for each negative sample independently such that the loss-augmented score is maximized (step 3-6 of Algorithm 2). The overall complexity of the above algorithm is $O(n^2)$ where n is the number of samples and it can be shown to provide the optimal ranking $\mathbf{Y}$ that maximizes the AP loss augmented score. We refer the interested reader to [2] for details.

C. Optimization for Latent SSVM

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:

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

The above problem belongs to a special class of non-convex optimization problems called difference-of-convex programs. Specifically, its feasible region can be viewed as the difference of two convex sets. Thus, we can employ a concave-convex procedure (CCCP) [3] to obtain an approximate solution, as described in Algorithm 3. Briefly, the CCCP algorithm starts with an initial set of parameters and iterates over two steps until convergence. In the first step, it fixes the parameters and finds the best set of additional annotations $\hat{\mathbf{H}}_P^*$ of the positives samples for the ground-truth output $\mathbf{Y}^*$. In the second step, it fixes the additional annotations $\hat{\mathbf{H}}_P^*$ and updates the parameters by solving the resulting convex optimization problem.

In more detail, the imputation of the additional annotations for the positive samples requires us to minimize the slack ξ , which is carried out by solving the following problem:

$$\begin{aligned} \argmin_{\hat{\mathbf{H}}_P} \{ & \Delta(\mathbf{Y}^*, \mathbf{Y}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\hat{\mathbf{H}}_P, \hat{\mathbf{H}}_N\}) \\ & + \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) \}. \end{aligned} \quad (4)$$

As the loss term $\Delta(\mathbf{Y}^*, \mathbf{Y})$ and score for incorrect ranking $\Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\})$ are independent of $\hat{\mathbf{H}}_P$, problem (4) reduces to the following:

$$\hat{\mathbf{H}}_P^* = \argmax_{\hat{\mathbf{H}}_P} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\hat{\mathbf{H}}_P, \hat{\mathbf{H}}_N\}). \quad (5)$$

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).

Given the imputed latent variables $\hat{\mathbf{H}}_P^*$, the parameter can be updated using the 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:

$$\begin{aligned} \hat{\mathbf{Y}}, \hat{\mathbf{H}}_N, \mathbf{H}_P, \mathbf{H}_N = \\ \argmax_{\mathbf{Y}, \hat{\mathbf{H}}_N, \mathbf{H}_P, \mathbf{H}_N} \{ & \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ & - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\hat{\mathbf{H}}_P^*, \hat{\mathbf{H}}_N\}) + \Delta(\mathbf{Y}^*, \mathbf{Y}) \}. \end{aligned} \quad (6)$$

We refer to the above problem as loss-augmented inference (since it augments the score of the ranking with its AP loss).

Algorithm 3 *The CCCP algorithm for parameter estimation of latent AP-SSVM.*

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

- 1: $t \leftarrow 0$
 - 2: **repeat**
 - 3: Impute the latent variables by solving problem (5).
 - 4: Update $\mathbf{w}_{t+1}$ by fixing the latent variables to $\hat{\mathbf{H}}_P^*$ and solving the following convex problem,

$$\begin{aligned} \mathbf{w}_{t+1} = \argmin_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi, \\ \text{s.t. } \quad & \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\hat{\mathbf{H}}_P^*, \hat{\mathbf{H}}_N\}) - \\ & \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) \geq \\ & \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi, \forall \mathbf{Y}, \hat{\mathbf{H}}_N, \mathbf{H}_P, \mathbf{H}_N \end{aligned}$$
 - 5: $t \leftarrow t + 1$
 - 6: **until** Objective function cannot be decreased below tolerance ϵ
-

To summarize, CCCP requires us to solve two problems: problem (5), that is, output-consistent inference and problem (6), that is, loss-augmented inference. We now describe how both the problems can be solved efficiently to obtain an accurate set of parameters for latent SSVM.

Output-Consistent Inference. Problem (5) can be solved efficiently by independently choosing the additional annotation for each positive sample using the following criterion:

$$\mathbf{h}_i^* = \argmax_{\mathbf{h}_i} s_i(\mathbf{h}_i). \quad (7)$$

Loss-Augmented Inference. To the best of our knowledge, the second step of the CCCP algorithm cannot be solved optimally due to the lack of an efficient algorithm that computes the most violating constraint for the corresponding cutting plane algorithm. We now provide details of an approximate optimization algorithm for problem (6) that was used in our experiments.

An approximate solution of problem (6) can be obtained in two steps. In the first step we minimize the positive score $\mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\hat{\mathbf{H}}_P^*, \hat{\mathbf{H}}_N\})$ over $\hat{\mathbf{H}}_N$. Since, the positive score is independent of the loss, it decomposes over all negative samples and thus can be solved independently by choosing the additional annotation $\mathbf{h}_j$ for each negative

sample using the following criterion:

$$\hat{\mathbf{h}}_j = \operatorname{argmax}_{\mathbf{h}_j} s_j(\mathbf{h}_j). \quad (8)$$

The second step is to maximize the loss-augmented negative score. Note that loss-augmented inference requires us to not only obtain the ranking of the samples, but also the additional annotations of each sample. Let us denote the set of all the ranks that are occupied by a positive sample by $\mathcal{R}^+$. Similarly, the set of all the ranks that are occupied by a negative sample is denoted by $\mathcal{R}^-$. We first focus on the problem of finding the additional annotations of each sample for a given ranking, that is, for fixed sets $\mathcal{R}^+$ and $\mathcal{R}^-$. We will later describe how we approximately optimize over the rankings. Let us consider the task of obtaining the additional annotations for the positive samples. To solve this task, we construct a $|\mathcal{P}| \times |\mathcal{P}|$ matrix $\mathbf{S}_P$, such that $\mathbf{S}_P(i, a)$ is the score of assigning the sample $i \in \mathcal{P}$ to the rank $a \in \mathcal{R}^+$. Formally,

$$\mathbf{S}_P(i, a) = \max_{\mathbf{h}_i} c_a s_i(\mathbf{h}_i), \quad (9)$$

where

$$c_a = \max_{b \in \mathcal{R}^-} \delta(b < a) - \delta(b > a). \quad (10)$$

$\delta(\cdot)$ returns the number of positive, negative example pairs satisfying the condition in the argument. The best assignment of additional annotations for the positive samples can be found efficiently by applying the dynamic Hungarian algorithm [1] to the matrix $\mathbf{S}_P$. Similarly, to solve the task of obtaining the additional annotations for the negative samples, we construct a $|\mathcal{N}| \times |\mathcal{N}|$ matrix $\mathbf{S}_N$, such that $\mathbf{S}_N(j, a)$ is the score of assigning the sample $j \in \mathcal{N}$ to the rank $a \in \mathcal{R}^-$, that is,

$$\mathbf{S}_N(j, a) = \max_{\mathbf{h}_j} c_a s_j(\mathbf{h}_j), \quad (11)$$

where

$$c_a = \max_{b \in \mathcal{R}^+} \delta(b > a) - \delta(b < a). \quad (12)$$

Once again, the best assignment of additional annotations for the negative samples can be found efficiently by applying the dynamic Hungarian algorithm [1] to the matrix $\mathbf{S}_N$.

The above argument shows that, for a given ranking, the optimal assignment of additional annotations for both the positive and the negative samples can be obtained in a computationally feasible manner. However, the optimization over the ranking itself poses a difficult problem. In order to obtain an approximate solution, we employ the following greedy strategy, which is a natural extension of the algorithm proposed by Yue *et al.* [2] for the supervised learning case. We start with the perfect ranking, where all the positive samples are ranked higher than all the negative samples. Next, we consider shifting the highest negative rank up the

ranking while keeping all other ranks fixed. For each such ranking, we compute the value of the loss. Furthermore, we also compute the assignment of additional annotation to the ranks such that the score is maximized. We pick the ranking that maximizes the loss augmented score among all such rankings. Next, we consider shifting the second highest negative rank up the ranking, and find the ranking that maximizes the loss augmented negative score. We continue this procedure until we have considered shifting the lowest negative rank up the ranking.

Non-Optimality of Loss-Augmented Inference The above algorithm can be shown to be non-optimal using a counter-example. Table 1 describes one such example with two positive and negative samples each. Each sample has two possible values for additional annotation. The first row of the table represents the identifier of each sample. The second and third rows represent the score of the sample upon choosing first or second value of additional annotation respectively. The bottom row contains the label of each sample.

Sample ID	0	1	2	3
Score ₀	4	7	1	0
Score ₁	8	6	9	7
Label	-1	1	-1	1

Table 1. Counter-example with two positive and negative samples each. Each sample has two possible values for additional annotation. The first row of the table represents the identifier of each sample. The second and third rows represent the score of the sample upon choosing first or second value of additional annotation respectively. The bottom row contains the label of each sample.

We represent the ranking as a list of ordered pairs (a, b) . The first entry of the ordered pair is the Sample-ID and second entry is the index of additional annotation selected:

The global optimum for the example in Table 1 is the following ranking,

$$(0, 1); (2, 1); (3, 0); (1, 1),$$

which yields a loss augmented score of is 22.6.

We now compare the global optimum with the solution obtained using the greedy algorithm described above. We start with the perfect ranking, where all the positive samples are ranked higher than all the negative samples. For the example given in Table 1, we have the following initial ranking.

$$++--$$

The best assignment of additional annotations for the samples can be found by applying the dynamic Hungarian algorithm. We have the following assignments for the initial ranking,

$$(1, 0); (3, 1); (2, 0); (0, 0),$$

which yields a loss augmented score of 18.

Next, we consider shifting the highest negative rank up the ranking while keeping all other ranks fixed. Following is the corresponding ranking.

$$+ - + -$$

Following is the best assignment for this ranking, yielding a loss augmented score of 12.2,

$$(1, 0); (0, 0); (3, 0); (2, 0).$$

Next, we consider shifting the highest negative rank further up the ranking while keeping all other ranks fixed. Following is the corresponding ranking.

$$- + + -$$

The best assignment for this ranking is as follows, which yields a loss augmented score of 14.4.

$$(0, 1); (1, 0); (3, 0); (2, 0).$$

Out of the above 3 rankings, we pick the ranking that maximizes the loss augmented score. For this example, the following ranking maximizes the loss augmented score.

$$+ + - -$$

Next, we consider shifting the second highest negative rank up the ranking and find the ranking that maximizes the loss augmented negative score, that is, we find the best ranking among the following.

$$+ + - -$$

$$+ - + -$$

$$- + + -$$

From the above three rankings, the following ranking yields the maximum loss augmented score of 18.

$$+ + - -$$

However, the loss augmented score of the global optimum is 22.6. Therefore, using the counter-example we can prove that the solution from the greedy algorithm described above is sub-optimal.

D. Latent AP-SVM as an Upper Bound of AP Loss

Before deriving the proof for our proposition that the latent AP-SVM formulation provides a tighter upper bound on the AP loss compared to the standard latent SSVM formulation, it would be helpful to prove the following lemma.

Lemma 1. For a given value of the ranking matrix $\mathbf{Y}$ and the additional annotations of the negative samples $\mathbf{H}_N$, consider the following optimization problem:

$$\mathbf{H}_P^* = \underset{\mathbf{H}_P}{\operatorname{argmin}} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}), \quad (13)$$

where $\mathbf{Y}^*$ is the optimal ranking matrix. The above optimization problem can be solved optimally as follows regardless of the choice of $\mathbf{Y}$ and $\mathbf{H}_N$:

$$\mathbf{h}_i^* = \underset{\mathbf{h}_i}{\operatorname{argmax}} \mathbf{w}^\top \Phi_i(\mathbf{h}_i). \quad (14)$$

Proof. We use the following shorthand:

$$s_i(\mathbf{h}_i) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \mathbf{w}^\top \Phi_i(\mathbf{h}_i),$$

$$s_j(\mathbf{h}_j) = \frac{1}{|\mathcal{P}||\mathcal{N}|} \mathbf{w}^\top \Phi_j(\mathbf{h}_j). \quad (15)$$

Using the above shorthand, we can rewrite the expression in the RHS of problem (13) as follows:

$$\begin{aligned} & \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ &= \sum_{i \in \mathcal{P}} s_i(\mathbf{h}_i) \left(\sum_{j \in \mathcal{N}} (Y_{ij} - Y_{ij}^*) \right) + \sum_{j \in \mathcal{N}} s_j(\mathbf{h}_j) \left(\sum_{i \in \mathcal{P}} (Y_{ij}^* - Y_{ij}) \right). \end{aligned} \quad (16)$$

By definition $\sum_{j \in \mathcal{N}} Y_{ij}^* \geq \sum_{j \in \mathcal{N}} Y_{ij}$ since $Y_{ij}^* = 1$ for all $i \in \mathcal{P}$ and $j \in \mathcal{N}$. Thus, the coefficient of the score term $s_i(\mathbf{h}_i)$ is negative for all positive samples. Therefore, in order to minimize problem (13) over all possible choices of $\mathbf{H}_P$, we should maximize the score $s_i(\mathbf{h}_i)$ for each positive sample. This is exactly the solution proposed in equation (14), which completes the proof. $\square$

We are now ready to prove the following proposition.

Proposition 1. The latent AP-SVM formulation provides a tighter upper bound on the AP loss compared to the standard latent SSVM formulation.

Proof. We compare the optimization problems corresponding to the standard latent SSVM formulation and the latent AP-SVM formulation. The parameters of latent SSVM are estimated by solving the following problem:

$$\begin{aligned} & \min_{\mathbf{w}} \frac{1}{2} \|\mathbf{w}\|^2 + C\xi, \\ & \text{s.t. } \forall \mathbf{Y}, \hat{\mathbf{H}}_N, \mathbf{H}_P, \mathbf{H}_N : \\ & \max_{\mathbf{H}_P} \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \hat{\mathbf{H}}_P, \hat{\mathbf{H}}_N) \} \\ & - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \mathbf{H}_P, \mathbf{H}_N) \geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi. \end{aligned} \quad (17)$$

The parameters of latent AP-SVM are estimated by solving the following problem:

$$\begin{aligned} \min_{\mathbf{w}} \quad & \frac{1}{2} \|\mathbf{w}\|^2 + C\xi, \\ \text{s.t.} \quad & \forall \mathbf{Y}, \mathbf{H}_N : \max_{\mathbf{H}_P} \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ & - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) \} \geq \Delta(\mathbf{Y}^*, \mathbf{Y}) - \xi. \end{aligned} \quad (18)$$

Note that the constraints specified in problem (18) are a subset of the constraints specified in problem (17). In other words, the feasible region of problem (18) is a superset of the feasible region of problem (17). It follows that the optimal objective function value of problem (18) is guaranteed to be less than or equal to the optimal objective function value of problem (17). In other words, minimizing problem (18) provides a quantity that is guaranteed to be less than or equal to the regularized upper bound on the AP loss that is optimized by the standard latent SSVM. Thus, in order to prove proposition 1 it is sufficient to show that problem (18) minimizes a valid regularized upper bound on the AP loss.

In order to show that problem (18) minimizes a regularized upper bound on the AP loss, we introduce the following notation. Given a set of parameters $\mathbf{w}$ for the latent AP-SVM formulation, we denote the predicted additional annotations by $H(\mathbf{w}) = (\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N(\mathbf{w}))$. Recall that, for each sample $\mathbf{x}_i$, the additional annotation $\mathbf{h}_i(\mathbf{w})$ is predicted using the following rule:

$$\mathbf{h}_i(\mathbf{w}) = \operatorname{argmax}_{\mathbf{h}_i} \mathbf{w}^\top \phi_i(\mathbf{h}_i). \quad (19)$$

Similarly, we denote the predicted output (that is, the ranking matrix) as $\mathbf{Y}(\mathbf{w})$, which is obtained using the following rule:

$$\mathbf{Y}(\mathbf{w}) = \operatorname{argmax}_{\mathbf{Y}} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \mathbf{H}(\mathbf{w})). \quad (20)$$

Using the above notation, the AP loss over the training dataset $\Delta(\mathbf{Y}^*, \mathbf{Y}(\mathbf{w}))$ can be upper bounded as follows:

$$\begin{aligned} & \Delta(\mathbf{Y}^*, \mathbf{Y}(\mathbf{w})) \\ = & \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}(\mathbf{w}), \{\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N(\mathbf{w})\}) \\ & + \Delta(\mathbf{Y}^*, \mathbf{Y}(\mathbf{w})) \\ & - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}(\mathbf{w}), \{\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N(\mathbf{w})\}) \end{aligned} \quad (21)$$

$$\begin{aligned} \leq & \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}(\mathbf{w}), \{\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N(\mathbf{w})\}) \\ & + \Delta(\mathbf{Y}^*, \mathbf{Y}(\mathbf{w})) \\ & - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N(\mathbf{w})\}) \end{aligned} \quad (22)$$

$$\begin{aligned} \leq & \max_{\mathbf{Y}, \mathbf{H}_N} \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N\}) \\ & + \Delta(\mathbf{Y}^*, \mathbf{Y}) \\ & - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P(\mathbf{w}), \mathbf{H}_N\}) \}. \end{aligned} \quad (23)$$

Expression (22) follows from the fact that the score for optimal ranking $\mathbf{Y}^*$ must be less than or equal to the score

for the predicted ranking $\mathbf{Y}(\mathbf{w})$ (since the predicted ranking is obtained by maximizing the score as shown in equation (20)). Expression (23) follows from the fact that instead of using the prediction, we maximize over all possible rankings and additional annotations of the negative samples.

Note that expression (23) still contains the predicted value of the additional annotations of the positive samples. In order to further simplify the upper bound, we make use of lemma (1), which implies that for any value of $\mathbf{Y}$ and $\mathbf{H}_N$, the following holds true:

$$\begin{aligned} \mathbf{H}_P(\mathbf{w}) = \operatorname{argmin}_{\mathbf{H}_P} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \\ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}), \end{aligned} \quad (24)$$

since the predicted value of $\mathbf{H}_P(\mathbf{w})$ is exactly equal to that specified by equation (14). Therefore, it follows that

$$\begin{aligned} \mathbf{H}_P(\mathbf{w}) = \operatorname{argmin}_{\mathbf{H}_P} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \\ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) + \Delta(\mathbf{Y}^*, \mathbf{Y}), \end{aligned} \quad (25)$$

since $\Delta(\mathbf{Y}^*, \mathbf{Y})$ is independent of $\mathbf{H}_P$. For example, consider the action classification task, where $\mathbf{Y}$ denotes the ranking of the images and $\mathbf{H}_P$ is the bounding boxes in the positive samples. Regardless of the choice of the positive samples, the loss is computed based on the ranking $\mathbf{Y}$ and not the bounding boxes $\mathbf{H}_P$. Using the fact that the above equation holds true for any choice of $\mathbf{Y}$ and $\mathbf{H}_N$, expression (23) can be simplified as

$$\begin{aligned} \min_{\mathbf{H}_P} \max_{\mathbf{Y}, \mathbf{H}_N} \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ + \Delta(\mathbf{Y}^*, \mathbf{Y}) \\ - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \}. \end{aligned} \quad (26)$$

We note that this is exactly the value of the slack variable ξ in problem (18). This proves that problem (18) minimizes a valid regularized upper bound of the AP loss, which in turn proves the proposition. $\square$

E. Efficient inference for AP-SVM

Output-Consistent Inference. Output-consistent inference requires us to solve the following problem:

$$\begin{aligned} \min_{\mathbf{H}_P} \max_{\mathbf{Y}, \mathbf{H}_N} \{ \Delta(\mathbf{Y}^*, \mathbf{Y}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \\ + \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) \}. \end{aligned} \quad (27)$$

Although problem (27) 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 (27) can be solved efficiently by independently choosing the latent variable for each positive sample using the following criterion:

$$\mathbf{h}_i^* = \operatorname{argmax}_{\mathbf{h}_i} \mathbf{w}^\top \Phi_i(\mathbf{h}_i), \forall i \in \mathcal{P}. \quad (28)$$

Proof. For any given value of $\mathbf{Y}$ and $\mathbf{H}_N$, lemma (1) proves the following:

$$\mathbf{H}_P^* = \underset{\mathbf{H}_P}{\operatorname{argmin}} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}), \quad (29)$$

where $\mathbf{H}_P^*$ is computed as suggested in the above proposition, that is, by maximizing the scores of positive samples independently over their choice of additional annotations. Therefore, it follows that

$$\mathbf{H}_P^* = \underset{\mathbf{H}_P}{\operatorname{argmin}} \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) + \Delta(\mathbf{Y}^*, \mathbf{Y}), \quad (30)$$

since $\Delta(\mathbf{Y}^*, \mathbf{Y})$ is independent of $\mathbf{H}_P$. For example, consider the action classification task, where $\mathbf{Y}$ denotes the ranking of the images and $\mathbf{H}_P$ is the bounding boxes in the positive samples. Regardless of the choice of the positive samples, the loss is computed based on the ranking $\mathbf{Y}$ and not the bounding boxes $\mathbf{H}_P$. Using the fact that the above equation holds true for any choice of $\mathbf{Y}$ and $\mathbf{H}_N$, it follows that

$$\mathbf{H}_P^* = \underset{\mathbf{H}_P}{\operatorname{argmin}} \max_{\mathbf{Y}, \mathbf{H}_N} \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P, \mathbf{H}_N\}) + \Delta(\mathbf{Y}^*, \mathbf{Y}) - \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P, \mathbf{H}_N\}) \}. \quad (31)$$

This proves the proposition. $\square$

Loss-Augmented Inference. The most violated constraint, is found by solving the following problem:

$$\hat{\mathbf{Y}}, \hat{\mathbf{H}}_N = \underset{\mathbf{Y}, \mathbf{H}_N}{\operatorname{argmax}} \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}, \{\mathbf{H}_P^*, \mathbf{H}_N\}) - \{ \mathbf{w}^\top \Psi(\mathbf{X}, \mathbf{Y}^*, \{\mathbf{H}_P^*, \mathbf{H}_N\}) + \Delta(\mathbf{Y}^*, \mathbf{Y}) \}. \quad (32)$$

Yue *et al.* [2] proposed an efficient method for loss-augmented inference over the output ranking $\mathbf{Y}$. However, in problem (32) we need to maximize over an additional set of variables $\mathbf{H}_N$. The following proposition allows us to perform the joint optimization efficiently.

Proposition 3. Problem (32) can be solved by first maximizing over $\mathbf{H}_N$ using the criterion, $\mathbf{h}_j = \operatorname{argmax}_{\mathbf{h}_j} s_j(\mathbf{h}_j)$.

Proof. Problem (32) can be written as

$$\underset{\mathbf{Y}, \mathbf{H}_N}{\operatorname{argmax}} \{ \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} (\mathbf{Y}_{ij}^* - \mathbf{Y}_{ij}) s_j(\mathbf{h}_j) + \sum_{i \in \mathcal{P}} \sum_{j \in \mathcal{N}} (\mathbf{Y}_{ij} - \mathbf{Y}_{ij}^*) s_i(\mathbf{h}_i^*) + \Delta(\mathbf{Y}^*, \mathbf{Y}) \}. \quad (33)$$

By definition, $\mathbf{Y}_{ij}^* = 1$ for all $i \in \mathcal{P}, j \in \mathcal{N}$. Thus, the coefficient of the score term $s_j(\mathbf{h}_j)$ is non-negative for all negative samples regardless of the value of $\mathbf{Y}$. Therefore, problem (32) can be maximized first over $\mathbf{H}_N$ by maximizing the score of each negative sample independently. This proves the proposition. $\square$

F. Additional Results on Character Recognition in Natural Images

Figure 1 shows the best AP values for all the 4 classes where the performance of latent AP-SVM is statistically different from that of latent SSVM (using paired t-test with p-value less than 0.05). Latent SSVM provides statistically significant improvements over latent AP-SVM for only 1 class. In contrast, latent AP-SVM improves the performance for 3 classes. In terms of the mean AP value, latent AP-SVM provides an improvement of 4.4% for the 4 classes shown in figure 1.

Figure 2 shows the AP values for the 4 statistically significant characters on the ‘test’ set. Similar to the cross-validation results, latent AP-SVM outperforms latent SSVM for 3 out of the 4 classes. In terms of the mean AP value, latent AP-SVM provides an improvement of 3.1% for the 4 classes shown in figure 2.

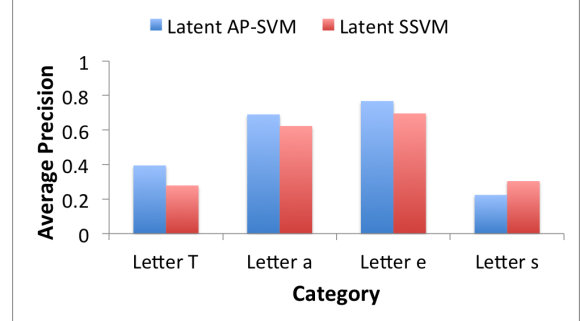


Figure 1. The best average precision values for the 4 statistically significant character classes obtained during 5-fold cross validation on the ‘trainval’ set of the IIIT 5K-WORD dataset. The x-axis corresponds to the characters. The y axis corresponds to the average precision. Latent AP-SVM provides statistically significant improvements over latent SSVM for 3 out of the 4 characters.

Figure 3 and figure 4 show the AP values for all 22 characters obtained during cross validation and testing respectively. In terms of the mean AP value over all 22 classes, latent AP-SVM provides an improvement of 3.2% and 2.7% over latent SVM during cross validation and testing respectively. Similarly, latent AP-SVM provides an improvement of 2.7% and 2.5% over latent SSVM during cross validation and testing respectively.

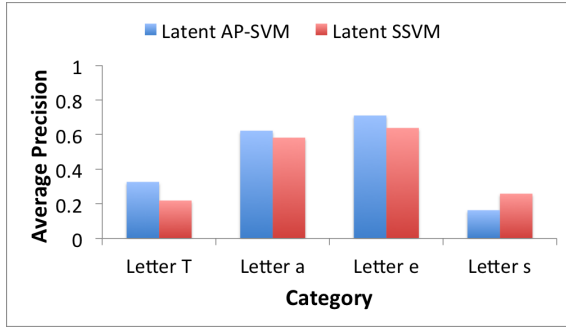


Figure 2. The average precision values for the 4 statistically significant characters obtained on the ‘test’ set of the IIIT 5K-WORD dataset. The x-axis corresponds to the character categories. The y axis corresponds to the average precision.

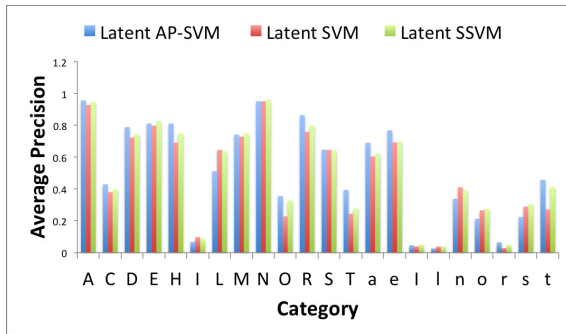


Figure 3. The best average precision values for all 22 character classes obtained during 5-fold cross validation on the ‘trainval’ set of the IIIT 5K-WORD dataset. The x-axis corresponds to the characters. The y axis corresponds to the average precision.

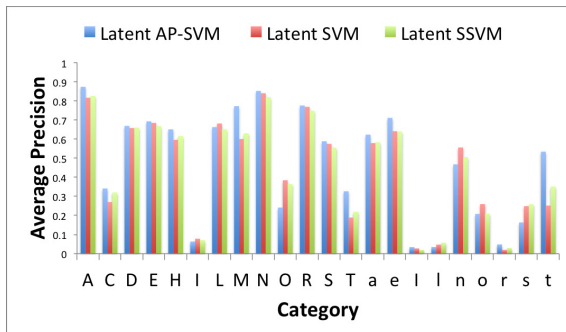


Figure 4. The average precision values for all 22 characters obtained on the ‘test’ set of the IIIT 5K-WORD dataset. The x-axis corresponds to the character categories. The y axis corresponds to the average precision.

References

- [1] G. A. Korsah, A. T. Stentz, and M. B. Dias. The dynamic hungarian algorithm for the assignment problem with changing costs. Technical report, Robotics Institute, 2007. 3
- [2] Y. Yue, T. Finley, F. Radlinski, and T. Joachims. A support

vector method for optimizing average precision. In *SIGIR*, 2007. 1, 3, 6

- [3] A. Yuille and A. Rangarajan. The concave-convex procedure. *Neural Computation*, 2003. 2