Sequential Labeling With Structural SVM Under an Average Precision Loss

Guopeng Zhang and Massimo Piccardi

Global Big Data Technologies Centre, University of Technology Sydney, NSW, Australia
Guopeng.Zhang@student.uts.edu.au, Massimo.Piccardi@uts.edu.au

Abstract. The average precision (AP) is an important and widely-adopted performance measure for information retrieval and classification systems. However, owing to its relatively complex formulation, very few approaches have been proposed to learn a classifier by maximising its average precision over a given training set. Moreover, most of the existing work is restricted to i.i.d. data and does not extend to sequential data. For this reason, we herewith propose a structural SVM learning algorithm for sequential labeling that maximises an average precision measure. A further contribution of this paper is an algorithm that computes the average precision of a sequential classifier at test time, making it possible to assess sequential labeling under this measure. Experimental results over challenging datasets which depict human actions in kitchen scenarios (i.e., TUM Kitchen and CMU Multimodal Activity) show that the proposed approach leads to an average precision improvement of up to 4.2 and 5.7 percentage points against the runner-up, respectively.

Keywords: Sequential labeling · structural SVM · average precision · loss-augmented inference.

1 Introduction and Related Work

Choosing appropriate performance measures plays an important role in developing effective information retrieval and classification systems. Common figures include the false positive and false negative rates, the precision and recall, and the F-measure which can all assess the accuracy of a prediction by comparing the predicted labels with given ground-truth labels. However, in applications such as information retrieval, it is often important to assess not only the accuracy of the predicted labels, but also that of a complete ranking of the samples. In classification, too, it is often preferable to evaluate the prediction accuracy at various trade-offs of precision and recall, to ensure coverage of multiple operating points. For both these needs, the average precision (a discretised version of the area under the precision-recall curve) offers a very informative performance measure.

Amongst the various flavours of classification, sequential labeling, or tagging, refers to the classification of each of the measurements in a sequence. It is a very important task in a variety of fields including video analysis, bioinformatics, financial time series and natural language processing [8]. Unlike the classification of independent samples,

the typical sequential labeling algorithms such as Viterbi (including their n-best versions [7]) do not provide multiple predictions at varying trade-offs of precision and recall, and therefore the computation of their average precision is not trivial.

In the literature, a number of papers have addressed the average precision as a performance measure in the case of independent samples. For instance, [5] has studied the statistical behaviour of the average precision in the presence of relevance judgements. Yilmaz and Aslam in [15] have proposed an approximation of the average precision in retrieval systems with incomplete and imperfect judgements. Morgan *et al*. in [6] have proposed an algorithm for learning the weights of a search query with maximum average precision. Notably, Joachims *et al*. in [16] have proposed a learning algorithm that can efficiently train a support vector machine (SVM) under an average precision loss. However, all this work only considers independent and identically distributed (i.i.d.) samples, while very little work to date has addressed the average precision in sequential labeling and structured prediction. In [9], Rosenfeld *et al*. have proposed an algorithm for training structural SVM under the average precision loss. However, their algorithm assumes that the structured output variables can be ranked in a total order relationship which is generally restrictive.

For the above reasons, we propose a training algorithm that can train structural SVM for sequential labeling under an average precision loss. Our assumptions are very general and do not require ranking of the output space. The core component of our training algorithm is an inference procedure that returns sequential predictions at multiple levels of recall. The same inference procedure can also be used at test time, making it possible to evaluate the average precision of sequential labeling algorithms and to compare it with that of i.i.d. classifiers.

Experiments have been conducted over two challenging sequential datasets: the TUM Kitchen and the CMU-MMAC activity datasets [11, 1]. The results, reported in terms of average precision, show that the proposed method remarkably outperforms other performing classifiers such as standard SVM and structural SVM trained with conventional losses.

2 Background

2.1 Average Precision

The average precision (AP) is a de-facto standard evaluation in the computer vision community since the popular PASCAL VOC challenges [2]. It is defined as the average of the precision at various levels of recall and is a discretised version of the area under the precision-recall curve (AUC). The AP is a very informative measure since it assesses the classification performance at different trade-offs of precision and recall, reflecting a variety of operating conditions. Its formal definition is:

$$AP = \frac{1}{R} \sum_{r} \; p_{@r} \tag{1}$$

where $p_{@r}$ is the precision at level of recall r, and R is the number of levels. The recall ranges between 0 and 1, typically in 0.1 steps. At its turn, the precision at a chosen value of recall, $p_{@r}$, is defined as:

$$p_{@r} = \frac{TP}{TP + FP} \quad s.t. \quad \frac{TP}{TP + FN} = r \tag{2}$$

where TP, FP and FN are the number of true positives, the number of false negatives and the number of false positives, respectively, computed from the classification contingency table of the predicted and ground-truth labels.

In general, the precision tends to decrease as r grows. However, it is not a monotonically non-increasing function of r. To ensure monotonicity of the summand, Everingham *et al*. in [3] modified the definition of the AP as:

$$AP = \frac{1}{R} \sum_{r} \max_{l=0 \ldots r} p_{@l} \tag{3}$$

This way of computing the average precision has become commonplace in the computer vision and machine learning communities and it is therefore adopted in our experiments. However, the algorithm we describe in Section 3 can be used interchangeably for either (1) or (3). Given that the AP is bounded between 0 and 1, a natural definition for an AP-based loss is $\Delta_{AP} = 1 - AP$.

2.2 Sequential labeling

Sequential labeling predicts a sequence of class labels, $y = (y_1, \ldots, y_t, \ldots, y_T)$, from a given measurement sequence, $x = (x_1, , \ldots, x_t, \ldots, x_T)$, where x_t is a feature vector at sequence position t and y_t is a corresponding discrete label, $y_t \in 1 \ldots M$. In many cases, it is not restrictive to assume that y_t is a binary label (1: positive class; 0: negative class), obtaining multi-class classification from a combination of binary classifiers. Therefore, in the following we focus on the binary case. The most widespread model for sequential labeling is the hidden Markov model (HMM) which is a probabilistic graphical model factorising the joint probability of the labels and the measurements. By restricting the model to the exponential family of distributions and expressing the probability in a logarithmic scale, the score of an HMM can be represented as a generalised linear model:

$$\begin{aligned} \ln p(x, y) \propto w^\top \phi(x, y) = w_{init}^\top f(y_1) + \\ + \sum_{t=2}^{T} w_{tran}^\top f(y_{t-1}, y_t) + \sum_{t=1}^{T} w_{em}^\top f(x_t, y_t) \end{aligned} \tag{4}$$

where w_{init} are the first-frame parameters, w_{tran} are the transition parameters, w_{em} are the emission parameters, and functions $f(y_1)$, $f(y_{t-1}, y_t)$ and $f(x_t, y_t)$ are arbitrary feature functions of their respective arguments. The inference problem for this model consists of determining the best class sequence for a given measurement sequence:

$$\bar{y} = \operatorname*{argmax}_{y} \; w^\top \phi(x, y) \tag{5}$$

This problem can be efficiently solved in $O(T)$ time by the well-known Viterbi algorithm operating in a logarithmic scale [8].

2.3 Structural SVM

SVM has been extended from independent (measurement, label) pairs to the prediction of structured labels, i.e. multiple labels that have mutual dependencies in the form of sequences, trees and graphs and that co-depend on multiple measurements [10, 12]. Given a set of N training instances $\{x^i, y^i\},\ i = 1, \ldots, N$, structural SVM finds the optimal model's parameter vector w by solving the following convex optimisation problem:

$$\begin{aligned} &\min_{w,\xi}\ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi^i \quad s.t. \\ &w^\top\phi(x^i, y^i) - w^\top\phi(x^i, y) \geq \Delta(y^i, y) - \xi^i, \\ &\qquad i = 1 \ldots N,\ \ \forall y \in \mathcal{Y} \end{aligned} \tag{6}$$

As usual, term $\sum_{i=1}^{N}\xi^i$ places an upper bound over the total training error, while term $\|w\|^2$ regularises the solution to encourage generalisation. Parameter C is an arbitrary, positive coefficient that balances these two terms. In the constraints, function $\phi(x, y)$ is a feature function that computes structured features from the pair $\{x, y\}$ such that $w^\top\phi(x, y)$ can assign a score to the pair. The constraint for labeling $y = y^i$ guarantees that $\xi^i \geq 0$, and $\Delta(y^i, y)$ is the chosen, arbitrary loss function.

The problem with Eq. (6) is that the size of the constraint set, $\mathcal{Y}$, is exponential in the number of of the output variables and it is therefore impossible to satisfy the full constraint set. However, [12] has shown that it is possible to find ϵ-correct solutions with a constraint subset of polynomial size, consisting of only the "most violated" constraint for each sample, i.e. the labeling with the highest sum of score and loss:

$$\begin{aligned} &\xi^i = \max_y(-w^\top\phi(x^i, y^i) + w^\top\phi(x^i, y) + \Delta(y^i, y)) \\ &\rightarrow \bar{y}^i = \operatorname*{argmax}_y(w^\top\phi(x^i, y) + \Delta(y^i, y)) \end{aligned} \tag{7}$$

This problem is commonly referred to as "loss-augmented inference" due to its resemblance to the usual inference of Eq. (5) and is the main step of structural SVM.

3 Training and Testing Sequential Labeling with the AP Loss

The loss functions used for training structural SVM commonly include the 0-1 loss and the Hamming loss. Under these losses, the loss-augmented inference can still be computed by a conventional Viterbi algorithm with adjusted weights. Instead, training with the average precision cannot be approached in the same way since it requires predicting either a ranking or multiple labelings. For this reason, we propose a different formulation of the structural SVM primal problem:

$$\begin{aligned}
&\min_{w,\xi} \ \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{N}\xi^i \quad s.t. \\
&w^\top\phi(x^i,y^i) - \frac{1}{R}\sum_r w^\top\phi(x^i,y^{[r]}) \\
&\quad \geq \Delta_{AP}(y^i,y^{[0]},\dots y^{[1]}) - \xi^i, \xi^i \geq 0, \ \ i = 1\dots N, \\
&\quad r = 0, 0.1, \dots 1, \ \ \forall y^{[0]}\dots y^{[1]} \in \mathcal{Y}_0 \times \dots \times \mathcal{Y}_1
\end{aligned} \tag{8}$$

The constraints in Eq. (8) state that the score assigned to the ground-truth labeling, y^i, must be greater than or equal to the average score of any set of R labelings at the appropriate levels of recall by at least their average precision loss. In this way, we retain the structural SVM principle of imposing a margin between the ground truth and the prediction that is equal to the chosen loss, while we constrain all the predictions at the prescribed levels of recall. At the same time, we cannot ensure that the hinge loss ξ^i is an upper bound for $\Delta_{AP}(y^i,y^{[0]},\dots y^{[1]})$, and therefore the minimisation of the loss over the training set is only heuristic.

For Eq. (8), the loss-augmented inference becomes:

$$\begin{aligned}
&\bar{y}^{[0]}\dots\bar{y}^{[1]} = \operatorname*{argmax}_{y^{[0]}\dots y^{[1]}} \left(\frac{1}{R}\sum_r w^\top\phi(x^i,y^{[r]}) + \Delta_{AP}(y^i,y^{[0]},\dots y^{[1]})\right) \\
&= \operatorname*{argmax}_{y^{[0]}\dots y^{[1]}} \left(\frac{1}{R}\sum_r w^\top\phi(x^i,y^{[r]}) + \frac{1}{R}\sum_r \Delta_{p@r}(y^i,y^{[r]})\right) \\
&= \operatorname*{argmax}_{y^{[0]}}(w^\top\phi(x^i,y^{[0]}) + \Delta_{p@0}(y^i,y^{[0]})), \dots \operatorname*{argmax}_{y^{[1]}}(w^\top\phi(x^i,y^{[1]}) + \Delta_{p@1}(y^i,y^{[1]}))
\end{aligned} \tag{9}$$

where we have made use of the definition of average precision from Eq. (1). Eq. (9) shows an important property: that the R most violating labelings can be found independently of each other using the precision loss at the required level of recall. This is the key property for the algorithm we propose in the following sub-section.

3.1 Inference and loss-augmented inference

Once the model is trained, testing it to report its AP requires, once again, the ability to produce a set of R predictions at the required levels of recall. Therefore, the key problems for both training and testing can be summed up, respectively, as:

$$\operatorname*{argmax}_{y^{[r]}} \ (w^\top\phi(x^i,y^{[r]}) + \Delta_{p@r}(y^i,y^{[r]})) \tag{10}$$

$$\operatorname*{argmax}_{y^{[r]}} \ w^\top\phi(x^i,y^{[r]}) \tag{11}$$

The algorithm we propose hereafter works interchangeably for both Eqs. (10) and (11), and also for the modified AP loss of Eq. (3). Given any ground-truth label sequence, y^i, the degrees of freedom of the precision loss are only the number of false positives, FP, and false negatives, FN. By making a prediction in left-to-right order along the sequence, the running values of FP and FN can only increment or remain unchanged. We can thus still approach the solution of Eq. (10) by dynamic programming, extending the state of a partial solution to include: a) the ground-truth label of the current frame, y_t, as in conventional Viterbi; b) the number of false positives, FP, in sub-sequence $y_{1:t}$; and c) the number of false negatives, FN, in sub-sequence $y_{1:t}$. We use notation $\psi(FP, FN, y_t)$ to indicate the $y_{1:t}$ sub-sequence with the highest score for the given extended state, and $s(\psi)$ for its score. The generic induction step is as follows: at any time step, t, a partial solution is obtained by extending two of the partial solutions of time $t-1$ with the current prediction, y_t, and correspondingly incrementing either FP or FN if the prediction is incorrect, or neither if correct. After the final time step, T, Eq. (10) is computed over the stored sequences and the argmax returned. Algorithm 1 describes the solution formally.

4 Experiments

The proposed approach has been evaluated on two challenging datasets of human activities, TUM Kitchen and CMU Multimodal Activity (CMU-MMAC). Descriptions and results for these two datasets are reported in the following sub-sections. The compared algorithms include: a) the proposed method based on the AP loss; b) structural SVM using the common 0-1 loss and Hamming loss, and c) a baseline offered by a standard SVM that classifies each frame separately. For SVM training, we have used constant $C = 0.1$ (based on a preliminary cross-validation), the RBF kernel (for non-linearity), and, for SSVM, convergence threshold $\epsilon = 0.01$ (default). For the AP loss, given the greater computational complexity of the loss-augmented inference (approximately quadratic for sequences with sparse positives), we decode each sequence in sub-sequences of 300 frames each. To develop the software, we have used the SVM^{struct} package and its MATLAB wrapper [4, 13]. All experiments have been performed on a PC with an Intel i7 2.4GHz CPU with 8 GB RAM.

4.1 Results on the TUM Kitchen dataset

The TUM Kitchen dataset is a collection of activity sequences recorded in a kitchen equipped with multiple sensors [11]. In the kitchen environment, various subjects were asked to set a table in different ways, performing 9 actions, *Reaching*, *TakingSomething*, *Carrying*, *LoweringAnObject*, *ReleasingGrasp*, *OpeningADoor*, *ClosingADoor*, *OpeningADrawer* and *ClosingADrawer*. For our experiments, we have chosen to use the motion capture data from the left and right hands. These data consist of 19 sequences for each hand, each ranging in length between $1,000$ and $6,000$ measurements. The first 6 sequences were used for training and the remaining for testing. Each measurement is a 45-D vector of 3D body joint locations. Fig. 1.a shows a scene from this dataset.

Table 1 reports the results for activity recognition from the left and right hand sequences. The table shows that the mean of the AP over the nine classes is the highest for

Algorithm 1 Algorithm for computing the loss-augmented inference of Eq. (10).

Input: w, $x = (x_1, \ldots, x_T)$, $y^g = (y^g_1, \ldots, y^g_T)$ (ground-truth labels), r
Output: $\bar{y}^{[r]}$
Initialize: $FP_{max} = FN_{max} = 0$

// FP, FN: running variables for the number of false positives and false negatives
// pos, neg: number of positives and negatives in y^g
// $\psi(invalidarg) = NULL, s(NULL) = -\infty$, *[ ] = string concatenation operator*

ψ = FindHighestScoringSequences(w, x, y^g);
$\bar{y}^{[r]}$ = FindMostViolatingLabeling(ψ, r);
return $\bar{y}^{[r]}$

---

function FindHighestScoringSequences(w, x, y^g)
// Finds all highest-scoring sequences for any combinations of FP and FN:

if $y^g_t = 0$
  $FP_{max} = FP_{max} + 1$
  for $FP = 0 : FP_{max}, FN = 0 : FN_{max}, t = 1 : T$
    $\psi(FP, FN, y_t = 0) =$
     $\text{argmax}(s([\psi(FP, FN, y_{t-1} = 0), 0]), s([\psi(FP, FN, y_{t-1} = 1), 0]))$
    $\psi(FP, FN, y_t = 1) =$
     $\text{argmax}(s([\psi(FP - 1, FN, y_{t-1} = 0), 1]), s([\psi(FP - 1, FN, y_{t-1} = 1), 1]))$
else
  $FN_{max} = FN_{max} + 1$
  for $FP = 0 : FP_{max}, FN = 0 : FN_{max}, t = 1 : T$
    $\psi(FP, FN, y_t = 0) =$
     $\text{argmax}(s([\psi(FP, FN - 1, y_{t-1} = 0), 0]), s([\psi(FP, FN - 1, y_{t-1} = 1), 0]))$
    $\psi(FP, FN, y_t = 1) =$
     $\text{argmax}(s([\psi(FP, FN, y_{t-1} = 0), 1]), s([\psi(FP, FN, y_{t-1} = 1), 1]))$
return ψ
end function

---

function FindMostViolatingLabeling(ψ, r)
// Finds the labeling maximising the sum of score and loss:

$FN^* = round(pos\ (1 - r))$ *// sets the desired recall level*

find $\text{argmax}_{\bar{y}^{[r]}}\ s(\bar{y}^{[r]})$ over $FP = 0 : neg, FN = FN^*$
 $\bar{y}^{[r]} = \text{argmax}_{\psi}$
 $[s(\psi(FP, FN^*, y_T = 0)) + \Delta_{p@r}(pos, FP, FN),$
 $s(\psi(FP, FN^*, y_T = 1)) + \Delta_{p@r}(pos, FP, FN)]$

// for Eq. (11), just remove $\Delta_{p@r}$
// for the modified AP loss of Eq. (3), set $FN = 0 : FN^*$

return $\bar{y}^{[r]}$
end function

---

the proposed technique, with an improvement of 4.2 percentage points over the runner-up for both the left and right hand sequences. In addition, the proposed technique reports the highest average precision in all the classes with the left hand sequences, and in 8 classes out of 9 with the right hand sequences. In addition, the average precision of the proposed technique is about double that of the standard SVM baseline that does not leverage sequentiality.

Table 1. Comparison of the average precision over the TUM Kitchen dataset. SVM: standard SVM baseline; 0-1 loss and Hamming loss: structural SVM with conventional loss functions; AP loss: proposed technique.

	Average precision (%)			
Left hand sequences	SVM	0-1 loss	Hamming loss	AP loss
Reaching	24.5	44.8	18.5	**50.1**
TakingSomething	31.1	79.7	20.0	**80.7**
LoweringAnObject	19.3	44.6	16.9	**49.9**
ReleasingGrasp	18.1	53.2	25.0	**54.4**
OpeningADoor	10.9	9.1	9.1	**15.5**
ClosingADoor	9.2	9.1	9.1	**11.5**
OpeningADrawer	10.5	14.8	11.8	**20.6**
ClosingADrawer	10.9	9.1	9.1	**15.5**
Carrying	62.3	75.6	51.9	**80.2**
Mean	21.9	37.8	19.0	**42.0**
Right hand sequences	SVM	0-1 loss	Hamming loss	AP loss
Reaching	18.0	65.5	18.3	**68.9**
TakingSomething	12.8	**91.6**	14.1	90.9
LoweringAnObject	13.7	43.1	15.1	**47.7**
ReleasingGrasp	17.9	40.8	18.8	**45.4**
OpeningADoor	29.1	68.5	16.3	**73.9**
ClosingADoor	13.2	36.4	15.6	**41.3**
OpeningADrawer	14.7	26.8	13.8	**30.2**
ClosingADrawer	12.3	30.7	13.0	**38.0**
Carrying	58.7	85.4	63.1	**89.9**
Mean	21.3	54.3	20.8	**58.5**

4.2 Results on the CMU Multimodal Activity dataset

The CMU Multimodal Activity (CMU-MMAC) dataset contains multimodal measurements of the activities of 55 subjects preparing 5 different recipes: "brownies", a salad, a pizza, a sandwich and scrambled eggs [1]. For our experiments, we have chosen to use the video clips of the 12 subjects preparing brownies from a dry mix box. The actions performed by the subjects are very realistic and are divided over 14 basic activities. The length of the 12 video clips ranges from $8,000$ to $20,000$ frames. For the experiments, we have used the first 8 videos for training and the remaining 4 for testing. For the feature vector of each frame, we have first extracted dense SIFT features at a 32-pixel step and used k-means with 32 clusters to generate a codebook. Then, the descriptors of each frame have been encoded into a $4,096$-D VLAD vector [14]. Fig. 1.b displays

(a) (b)

Fig. 1. Sample frames from (a) the TUM Kitchen dataset and (b) the CMU-MMAC dataset.

a scene from this dataset, showing that the kitchen environment and camera view are significantly different from TUM's.

Table 2 reports the results for activity recognition over this dataset. The table shows that the mean of the AP is the highest for the proposed technique, with an improvement of 5.7 percentage points over the runner-up. In addition, the proposed technique reports the highest average precision for 12 classes out of 14, and more than doubles the SVM baseline.

5 Conclusion

The average precision has become a reference evaluation measure for its ability to assess performance at multiple operating points. However, the typical sequential labeling algorithms such as Viterbi do not allow the computation of the average precision. For this reason, in this paper, we have proposed an inference procedure that infers a set of predictions at multiple levels of recall and allows measuring the average precision of a sequential classifier. In addition, we have proposed a structural SVM training algorithm for sequential labeling that minimises an average precision loss. Experiments conducted over two challenging activity datasets - TUM Kitchen and CMU-MMAC - have shown that the proposed approach significantly outperforms all of the other compared techniques and more than doubles the performance of a baseline. Moreover, while we have only focused on sequential labeling in this paper, the proposed approach could readily be employed for more general structures such as trees and graphs.

References

1. De la Torre, F., Hodgins, J.K., Montano, J., Valcarcel, S.: Detailed human data acquisition of kitchen activities: the CMU-multimodal activity database (CMU-MMAC). In: CHI 2009 Workshops. pp. 1–5 (2009)
2. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC2007) Results. http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html

Table 2. Comparison of the average precision over the CMU-MMAC dataset. SVM: standard SVM baseline; 0-1 loss and Hamming loss: structural SVM with conventional loss functions; AP loss: proposed technique.

	Average Precision (%)			
	SVM	0-1 loss	Hamming loss	AP loss
Close	9.9	**16.8**	9.0	16.2
Crack	11.4	23.1	8.3	**28.5**
None	30.9	46.6	29.8	**54.1**
Open	15.8	**33.3**	16.1	29.0
Pour	30.6	50.0	27.4	**61.4**
Put	13.1	27.8	11.4	**34.3**
Read	9.1	10.9	11.8	**16.5**
Spray	14.8	25.6	10.2	**28.4**
Stir	28.0	39.4	23.5	**45.5**
Switch-on	11.3	27.6	9.9	**32.9**
Take	22.5	47.1	19.8	**60.8**
Twist-off	10.1	25.5	7.9	**30.0**
Twist-on	9.8	19.0	8.1	**27.6**
Walk	10.4	23.2	9.6	**29.7**
Mean	16.3	29.7	14.5	**35.4**

3. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The Pascal visual object classes (VOC) challenge. IJCV 88(2), 303–338 (2010)
4. Joachims, T.: $\mathrm{SVM}^{\mathrm{struct}}$: Support vector machine for complex output 3.10 (2008), `http://www.cs.cornell.edu/people/tj/$svm_light$/$svm_struct$.html`
5. Kishida, K.: Property of Average Precision and Its Generalization: An Examination of Evaluation Indicator for Information Retrieval Experiments. NII technical report, National Institute of Informatics (2005)
6. Morgan, W., Greiff, W., Henderson, J.: Direct maximization of average precision by hill-climbing, with a comparison to a maximum entropy approach. In: HLT-NAACL 2004. pp. 93–96. HLT-NAACL-Short '04 (2004)
7. Nilsson, D., Goldberger, J.: Sequentially finding the n-best list in hidden markov models. In: IJCAI'01. pp. 1280–1285 (2001)
8. Rabiner, L.: A tutorial on hidden Markov models and selected applications in speech recognition. IEEE Proc. 77, 257–286 (1989)
9. Rosenfeld, N., Meshi, O., Tarlow, D., Globerson, A.: Learning structured models with the AUC loss and its generalizations. In: AISTATS 2014. pp. 841–849 (2014)
10. Taskar, B., Guestrin, C., Koller, D.: Max-margin Markov networks. In: NIPS. pp. 25–32 (2003)
11. Tenorth, M., Bandouch, J., Beetz, M.: The TUM Kitchen Data Set of Everyday Manipulation Activities for Motion Tracking and Action Recognition. In: IEEE International Workshop on Tracking Humans for the Evaluation of their Motion in Image Sequences (THEMIS), in conjunction with ICCV2009. pp. 1089–1096 (2009)
12. Tsochantaridis, I., Joachims, T., Hofmann, T., Altun, Y.: Large margin methods for structured and interdependent output variables. JMLR 6, 1453–1484 (2005)

13. Vedaldi, A.: A MATLAB wrapper of $\mathrm{SVM}^{\mathrm{struct}}$ (2011), http://www.vlfeat.org/~vedaldi/code/svm-struct-matlab.html
14. Vedaldi, A., Fulkerson, B.: VLFeat: An open and portable library of computer vision algorithms (2008), http://www.vlfeat.org/index.html
15. Yilmaz, E., Aslam, J.A.: Estimating average precision with incomplete and imperfect judgments. In: ACM CIKM '06. pp. 102–111 (2006)
16. Yue, Y., Finley, T., Radlinski, F., Joachims, T.: A support vector method for optimizing average precision. In: SIGIR. pp. 271–278 (2007)