A Ranking Part Model for Object Detection

Chaobo Sun and Xiaojie Wang

School of Computer, Beijing University of Posts and Telecommunications
{cbsun,xjwang}@bupt.edu.cn

Abstract. Object detection has long been considered a binary-classification problem, but this formulation ignores the relationship between examples. Deformable part models, which achieve great success in object detction, have the same problem. We use learning to rank methods to train better deformable part models, and formulates the optimization problem as a generalized convex concave problem. Experiments show that, using same features and similar part configurations, performance of detection by the ranking model outperforms original deformable part models on both INRIA pedestrians and Pascal VOC benchmarks.

Keywords: Object Detection, Deformable Part Model, Learning to Rank.

1 Introduction

Object detection is a task for localizing objects of specific categories, it has been playing a critical role in high-level image understanding. Previous models formulate object detection as a binary classification problem[9,17,13]. All candidate detections are judged a true object area or not. These detections can be sampled either by sliding windows on a feature pyramid[9,13], or from a shrunk space generated by objectness models[1,17].

Here comes the problem: Assuming that the feature of a detection candidate is $\boldsymbol{x}$, and with its label y. Classification models only focus on the relationship between $\boldsymbol{x}$ and y, while ignoring relationships between different $\boldsymbol{x}$. We argue that these types of relationships are also important: for example, if a $\boldsymbol{x_a}$ is better than $\boldsymbol{x_b}$(that is, candidate a with feature $\boldsymbol{x_a}$ have a higher overlap ratio with some objects than candidate b with feature $\boldsymbol{x_b}$), an ideal model should give a a higher score. Similarly, if two detections have close best overlap ratios, the model should give them close scores. Classification models fails to model these situations. As Figure 1 shows, we focus on "Why is a detection better than another" rather than "Why is a detection true".

This paper aims to overcome the shortcomings of previous object detection mentioned above. Our contributions are:

- We provide a ranking perspective on object detection. To search objects from candidate space, three types of information are available:item-wise, pair-wise and list-wise. Classification models use only item-wise information, while our model uses both pair-wise and list-wise information.

P. Fränti et al. (Eds.): S+SSPR 2014, LNCS 8621, pp. 414–423, 2014.

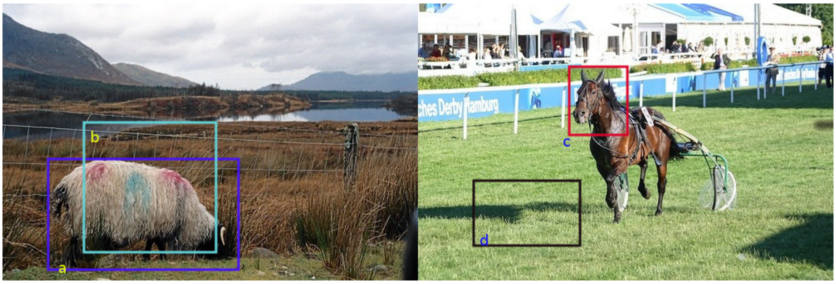

Fig. 1. In the image on the left-hand side, detection a and b are both true detections, but a is definitely a better detection than b; In the image on the right-hand side, c and d are both false detections whose overlap ratios with ground truth objects are below 0.5, but still we can tell c is better than d

- We propose a new objective function based on learning to rank theory, and apply it on deformable part models[13]. The objective function is a variant of LambdaRank[6].
- We formulate the optimization problem as a generalized-CCCP problem, and solve it in a similar way as CCCP[18].

We have a brief review on background works in section 2. The details of ranking formulation for object detction are discussed in section 3.2. In section 3.3 we describe our objective function. Section 4 shows the procedure of optimizing the objective function. Section 5 shows the results on well-known object detection datasets. In section 6 we conclude our work and discuss possible improvements.

2 Related Work

Research on generic object detection is originating from person detection[9]. From then on sliding-window methods with HOG pyramids have been a main stream on object detection. For every category of objects, sliding-window build a set of templates to represent all its poses. During training, cropped objects and backgrounds are extracted to train the template. During detecting, a matching score is computed at every position in the feature space, then the position with scores above a threshold is considered to be an object position[9].

Deformable part models(DPM)[13] have greatly pushed the research on object detection. As a variation of sliding windows, DPMs establish a set of hierarchical templates for every category of objects. Each template is organized into a root and its parts. Not only the appearance(vision features) of roots and parts, but also the parts' relative positions(structural features) to the root are taken into consider, so that DPMs can tolerate a certain degree of deformation.

Our work is mainly based on DPMs. We follow the definition of hierarchical templates, but improve the training procedure. Unlike an equivalent conversion from latent svm to latent struct svm[20], we use a totally new objective function based on the theory of learning to rank, and adopt it suitable for object detction.

We notice that there are several works on strong supervised models[4,2] that need additional annotations for parts. Additional annotations may help but reduces the difficulty of detection task. Our model can be applied on such models easily.

The work from Balschko[3] uses ranking svm[15] to model object detection, it is similar to our work in the sense that we both want to capture the essence why a detection is better than another. But there are significant differences: they use ranking to handle unlabeled data rather than take object detection as a ranking problem; they do not model the latent variables while we do, they use a svm-style objective function while we use a cross entropy style one, which is more flexible for adding list-wise information.

Learning to rank methods use cross entropy to measure distribution diverge between empirical probability and model probability[5]. LambdaRank[6] modifies the form of objective function by interpolating information retrieval measures. We adopt LambdaRank for more efficient computing in object detection.

During optimization of the cross-entropy style objective function, we find it an ensemble of Convex Concave Problems[18]. We also notice that finding a convex lower bound for the concave part of every CCCP would make the whole problem convex, so the two-stage optimization for original CCCP is suitable for our new problem.

3 Model

3.1 Deformable Part Models: A Review

Before introducing our model, let us have a brief review on deformable part models.

An object detection model defines a score function for detections. The function gives a confidence on the detections. Let x be the position of detections, I be the image on which detection is performed, a one-layer linear score function is defined as:

$$f(x, I; \omega, b) = \omega \cdot H_I(x) + b \tag{1}$$

where H_I is the feature pyramid, and $H_i(x)$ is the features covered by x. b is a bias.

DPMs introduce parts into original flat templates. Because locations of parts are un-observed variables, they may take any position in the image. Let position of parts be z, then the score function of DPM is simply a maximum of all possible zs:

$$f(x, I; \omega) = \max_z g(x, z, I; \omega) \tag{2}$$

The function g, which is a score function for joint x and z, is defined as:

$$g(x, z, I; \omega) = \omega_0^a \cdot H_I(x) + \sum_{k=1}^{K} [\omega_k^a \cdot H_I(z_k) - \omega_k^d \cdot d(x, z_i, v_i)] + \omega^b \tag{3}$$

Where $d(x, z_i, v_i)$ is the deformation function for z_i relative to x. v_is are ideal anchors for the ith part, they could be defined either heuristically[13] or by pre-defined rules[19]. We use ω to represent all parameters: ω^a for parameters of appearance, ω^d for parameters of deformation, and ω^b for bias. Note that $g(x, z, I; \omega)$ is linear function of ω

To train such functions, DPM then defines an svm-style loss function. Suppose we have n samples (x_i, I_i, y_i), where $y_i \in +1, -1$ representing whether x_i on I_i is a true detection or not.

$$L(\omega) = \frac{1}{2}\|\omega\|^2 + C \cdot \sum_{i=1}^{n} max(0, 1 - y_i f(x_i, I_i; \omega)) \tag{4}$$

3.2 Ranking Perspective on Object Detection

Classical learning to rank systems focus on selecting relevant items from a set of candidates. Object detection is similar to these models, if we interpret the searching space of object detection as a set of candidates. Following the way of Pascal VOC evaluation[11], the set of candidates are all positions of all images in a dataset. The aim of object detection is then selecting candidates that have more overlap ratios with ground truth objects.

In information retrieval systems, when modeling the relationship of some samples x and their corresponding labels y, there are three types of information:

- item-wise information, the direct relationship between x and y.
- pair-wise information, the relationship between a paired (x_i, x_j).
- list-wise information, the importance of x's position in the ordered list.

The key difficulties for applying ranking models on object detection is its large space of candidates. All rectangles in images are candidates to rank. Sliding window methods largely reduce the number of candidates by making the constraint that all candidates should be in certain sizes[9], while in recent years there are several useful technologies directly aiming to shrink the space of candidates[17,8].

We use a sliding window way to generate candidates, but it is very convenient to apply our model on a shrunk space of candidates.

3.3 Ranking DPM

Different from original DPM, we do not generate detections with their labels, we organize samples into a list of pairs. Instead of representing the list explicitly, we use a set of pairs, J, to represent the list. Every pair (i, j) in the set J means that detection x_i has a higher overlap ratio than x_j with some ground truth objects. Then the ordered list defined by J is strict partially ordered[16], and contains sufficient information of relationships between examples.

We define a simple empirical distribution on every pair (i, j) in J:

$$\bar{\mathrm{P}}_{\mathrm{ij}} \equiv \begin{cases} 1, & (i, j) \in J \\ 0, & (j, i) \in J \end{cases} \tag{5}$$

The empirical probability is a statistical measure of the pair-wise information in training datasets. During the train stage of our model, the score function is applied on each detction, and the score outputted would also generate a model distribution. We define it in a form of sigmoid function:

$$P_{ij} \equiv \frac{1}{1 + e^{-\sigma(f_i - f_j)}} \tag{6}$$

For simplicity, we use f_i to denote $f(x_i, I_i; \omega)$.

The two distributions should be as close as possible. We use cross entropy to measure the divergence of them:

$$C_{ij} = -\bar{P}_{ij} log(P_{ij}) - (1 - \bar{P}_{ij}) log(1 - P_{ij}) \tag{7}$$

Combining Eq.(5) ,(6) and (7), we got:

$$C_{i,j} = log(1 + e^{-\sigma(f_i - f_j)}) \tag{8}$$

To some extent, Eq.(8) can be a loss function on a single pair. Before summing up the C_{ij}s, it is necessary to examine how important the pair is in the whole strict ordered list. Note that C_{ij} is an increasing function of $f_i - f_j$, and an obvious fact is that for a pair of detections, they should be scored more discriminatively when they have bigger differences on overlap ratios. So it is reasonable to give the pairs weights according to the differences of overlap ratios.

This configuration is similar to [10], while we replace the changes of information retrieval measures with the differences of overlap ratios. As we discussed above, differences of overlap ratios plays a role similar to the changes of information retrieval measures, they both denote the importance of the binary relationship within the whole list.

Then we have the following loss function:

$$L(\omega) = \alpha \cdot \|\omega\|^2 + \sum_{(i,j) \in J} log(1 + e^{-\sigma(f_i - f_j)})(ov_i - ov_j) \tag{9}$$

α is a factor for regularization item. Note that x_i is a better detection than x_j, so the weight factor $ov_i - ov_j$ is always positive.

4 Optimization

4.1 Generalized CCCP

The loss function for every pair is not convex, but it is semi-convex in the sense that, the loss function is convex under specific constraints of f_i.

We have the following lemma:

Lemma 1. *if $f(x)$ and $g(x)$ are both convex, $g(x)$ is non-decreasing, then $g(f(x))$ is convex.*

With this lemma, we can prove that:

Theorem 1. *If f_i is concave and f_j is convex, Eq.*(9) *is convex.*

Proof $log(1 + e^{\sigma x})$ is convex and non-decreasing, then the loss function is convex if $f_j - f_i$ is convex. On the other hand, if f_i is concave, then $-f_i$ is convex, and then we get a convex $f_j - f_i$.

Recall that f is maximum of some linear functions, and therefore is convex. So, it is f_i that make Eq.(9) non-convex. But if we find a *concave* lower bound for f_i, the loss function is convex. As suggested in [18], we can obtain the concave function by fixing f_i with its best latent variable(part locations):

$$\begin{aligned} h_i &= g(I_i, x_i, z_i^*; \omega) \\ z_i^* &= \operatorname*{argmax}_z g(I_i, x_i, z; \omega) \end{aligned} \tag{10}$$

Then the linear function h_i is both convex and concave, and thus is a concave lower bound for f_i, and the loss function is convex if we replace f_i with h_i.

As we show above, $f_j - f_i$ is a Convex-Concave Problem(CCCP)[18], and the whole loss function is a so-called "Generalized CCCP".

4.2 Optimization Procedure

To solve Generalized CCCP, we follow a similar way to solve standard CCCP, which uses an iteration of two stages:

- *Latent Variable Finding.* In this stage, for every i that there exists some $(i, j) \in J$, we extract z_i^*, and calculate h_j based on z_i^*.
- *Optimization.* In this stage, we try to optimizing the convex problem $\alpha \cdot \|\omega\|^2 + \sum\limits_{(i,j)\in J} log(1 + e^{-\sigma(h_i - f_j)})(ov_i - ov_j)$.

It is worth noting that, for pairs (i, j) in the set of pairs J, some positive examples may be in the position of i in some pairs, while in the position of j in others. When scoring these examples, we have to calculate $h(x, I; \omega)$ and $f(x, I; \omega)$ simultaneously. For convenience of computing, we use $h(x, I; \omega)$ for scoring all examples from positive images.

In the latent variable finding stage, we use three subroutines:

detect_best denotes the procedure of finding z^* for ground truth boxes in I. Distance transform[12] is used in the max finding. All z^*s are extracted with their overlap ratios with ground truth boxes.

detect_hard denotes the procedure of detection on negative images, the positions with top scores are considered to be hard examples, and are selected. All examples are labeled with overlap ratio 0.

generate_pairs generates the set of pairs J using all examples, every pair with different overlap ratios are put into the set.

During the optimization stage, we use L-BFGS[7] as the loss function is derivable:

$$\nabla L(\omega) = 2\alpha \cdot \omega + \sum_{(i,j)\in J} \frac{-\sigma}{1 + e^{\sigma(h_i - f_j)}}(ov_i - ov_j)[\nabla h_i - \nabla f_j] \tag{11}$$

The training procedure is illustrated in Algorithm 1.

```
Data:
Positive Examples:$P = \{(I_1^P, B_1), \ldots, (I_n^P, B_n)\}$
Negative Examples:$N = \{I_1^N, \ldots, I_m^N\}$
Initial model parameters:$\omega^{old}$
Result: New model parameters:$\omega^{new}$
$\omega_0 := \omega^{old}$;
for $t$:=1 to $T$ do
    for $i$:=1 to $n$ do
        $F := detect\_best(I_i^P, \omega_{t-1})$;
        Add $F$ to $F_P$;
    end
    for $j$:=1 to $m$ do
        $F := detect\_hard(I_j^N, \omega_{t-1})$;
        Add $F$ to $F_N$;
    end
    $J := generate\_pairs(F_P, F_N)$;
    $\omega_t := l\_bfgs(J)$;
end
$\omega^{new} := \omega_T$
```

Algorithm 1. optimization

5 Experiments

We have evaluated our method on two well known datasets: INRIA pedestrians[9] and PASCAL VOC 2007[11]. Performance is measured in term of Average Precision (AP) according to the PASCAL VOC protocol[11].

We first initialize models with [14], and then apply our training procedure. To show whether the new objective function captures more information of training sets, we use the same features(an adopted version of HOG) suggested by [13].

5.1 INRIA Person

INRIA pedestrians dataset contains 1832 training images and 741 testing images [9]. Only persons are labeled with their bounding boxes. We evaluate our models and original DPM in Pascal VOC measures.

Figure 2 Shows the comparison of performances on INRIA dataset, our model promote the mAP measure from *0.8520* to *0.8571*. Simultaneously, our model gives a much smaller number(*1020*) of detections on test dataset compared to original DPM(*2952*). These results clearly shows that our model provides a more discriminative divide for object-related detections and backgrounds. The reduction of detection number would be very useful in practice.

But it is also worth noting that, it is not the main difference that our model provides a better divide. Our model has a higher precision almost at any recall value.

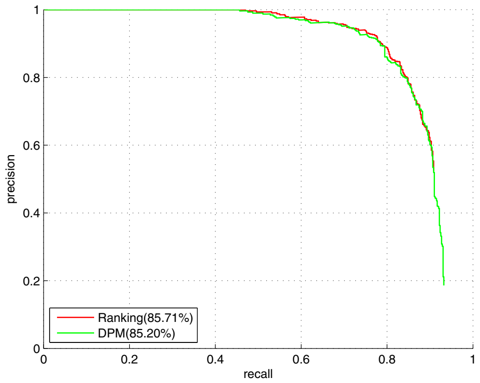

Fig. 2. Compared to original DPM, our model achieves better AP value while giving fewer candidates

5.2 Pascal VOC

Pascal VOC dataset is a more challenging benchmark. It contains 20 categories of objects, more complex backgrounds. Objects within each category have significant differences in appearances, scales and poses(for animals). And the numbers of objects in each category vary largely[11].

We evaluate our model and original DPM on the dataset. For convenience of comparison, we do not apply any post-processing technologies such as box predicting and context predicting.

Table 1 shows the results of our model(Ranking) and original DPM on Pascal VOC 2007. Our model outperforms original DPM on *bicycle*, *chair*,*dog*, *motorbike*, *sheep*, *train* and *tvmonitor*, while have a poorer performance on *bus* and *sofa*. Results on other categories are close.

The results show that in most cases, our model captures more characteristics of training data, and the characteristics are helpful or at least not harmful when applied on testing data. But still in some cases, the characteristics do play a role like noise.

Table 1. Evaluation results on Pascal VOC 2007, in Average Precision(%)

class	plane	bike	bird	boat	bottle	bus	car	cat	chair	cow	table	dog	horse	mbike	person	plant	sheep	sofa	train	tv	AVG
DPM	31.04	59.73	4.02	12.12	23.47	**50.55**	54.63	17.12	17.71	22.79	22.14	4.59	58.29	47.88	41.76	8.54	18.76	**35.86**	45.37	40.84	30.86
Ranking	31.04	**59.93**	4.02	12.12	**23.77**	50.51	54.63	17.12	**17.95**	22.80	22.14	**4.81**	58.29	**48.01**	**41.90**	8.54	**20.00**	35.50	**45.39**	**40.90**	30.96

6 Conclusion and Future Work

In this paper, we proposed a new modeling perspective on object detection: learning to rank. Following this perspective, we defined an ranking model based on information retrieval theory and DPM, and then formulated the optimization problem to a generalized CCCP. We evaluated our model on INRIA and Pascal VOC datasets, and performances on both benchmarks outperform original DPM which is based on latent svm.

According to our observation, the usefulness of the model is not directly linked with the shallow features of datasets, like the number of training examples or the ratio of positive examples and negative examples. The investigation of reasons are a main direction of our future work.

While ranking is useful for object detection, there are still differences of object detection with classical ranking problems: a much larger space of candidates. So it would be more useful to run the ranking model on a smaller space of candidates generated by objectness methods.

Acknowledgments. This work was partially supported by National Natural Science Foundation of China(Project No.61273365), National High Technology Research and Development Program of China(No. 2012AA011104) and discipline building plan in 111 base(No. B08004) and Engineering Research Center of Information Networks, Ministry of Education.

References

1. Alexe, B., Deselaers, T., Ferrari, V.: What is an object? In: 2010 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 73–80. IEEE (June 2010)
2. Azizpour, H., Laptev, I.: Object detection using strongly-supervised deformable part models. csc.kth.se (2012)
3. Blaschko, M.B., Vedaldi, A., Zisserman, A.: Simultaneous object detection and ranking with weak supervision. In: NIPS, vol. 1, p. 5 (2010)
4. Branson, S., Perona, P., Belongie, S.: Strong supervision from weak annotation: Interactive training of deformable part models. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1832–1839 (November 2011)
5. Burges, C., Shaked, T., Renshaw, E., Lazier, A., Deeds, M., Hamilton, N., Hullender, G.: Learning to rank using gradient descent. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 89–96. ACM (2005)
6. Burges, C.J., Ragno, R., Le. Learning, Q.V.: to rank with nonsmooth cost functions. In: NIPS, vol. 6, pp. 193–200 (2006)
7. Byrd, R.H., Lu, P., Nocedal, J., Zhu, C.: A limited memory algorithm for bound constrained optimization. SIAM Journal on Scientific Computing 16(5), 1190–1208 (1995)
8. Cheng, M.-M., Zhang, Z., Lin, W.-Y., Torr, P.H.S.: BING: Binarized normed gradients for objectness estimation at 300fps. In: IEEE CVPR (2014)
9. Dalal, N., Triggs, B., Europe, D.: Histograms of oriented gradients for human detection. In: IEEE Computer Society Conference on Computer Vision and Pattern Recognition, CVPR 2005, vol. 1, pp. 886–893. IEEE (2005)

10. Donmez, P., Svore, K.M., Burges, C.J.: On the local optimality of lambdarank. In: Proceedings of the 32nd International ACM SIGIR Conference on Research and Development in Information Retrieval, pp. 460–467. ACM (2009)
11. Everingham, M., Van Gool, L., Williams, C.K.I., Winn, J., Zisserman, A.: The PASCAL Visual Object Classes Challenge 2007 (VOC 2007) Results, http://www.pascal-network.org/challenges/VOC/voc2007/workshop/index.html
12. Felzenszwalb, P., Huttenlocher, D.: Distance transforms of sampled functions. Technical report, Cornell University (2004)
13. Felzenszwalb, P.F., Girshick, R.B., McAllester, D., Ramanan, D.: Object detection with discriminatively trained part-based models. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(9), 1627–1645 (2010)
14. Girshick, R.B., Felzenszwalb, P.F., McAllester, D.: Discriminatively trained deformable part models, release 5, http://people.cs.uchicago.edu/~rbg/latent-release5/
15. Joachims, T.: Optimizing search engines using clickthrough data. In: Proceedings of the Eighth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, pp. 133–142. ACM (2002)
16. Dan, A.: Simovici and Chabane Djeraba. In: Mathematical Tools for Data Mining. Springer (2008)
17. van de Sande, K.E.A., Uijlings, J.R.R., Gevers, T., Smeulders, A.W.M.: Segmentation as selective search for object recognition. In: 2011 IEEE International Conference on Computer Vision (ICCV), pp. 1879–1886. IEEE (2011)
18. Yuille, A.L., Rangarajan, A.: The concave-convex procedure (cccp). Advances in Neural Information Processing Systems 2, 1033–1040 (2002)
19. Zhu, L., Chen, Y., Yuille, A.: Learning a hierarchical deformable template for rapid deformable object parsing. IEEE Transactions on Pattern Analysis and Machine Intelligence 32(6), 1029–1043 (2010)
20. Zhu, L., Chen, Y., Yuille, A., Freeman, W.: Latent hierarchical structural learning for object detection. In: 2010 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), pp. 1062–1069. IEEE (2010)