Collective Activity Detection using Hinge-loss Markov Random Fields

Ben London, Sameh Khamis, Stephen H. Bach, Bert Huang, Lise Getoor, Larry Davis
University of Maryland
College Park, MD 20742
{blondon,sameh,bach,bert,getoor,lsdavis}@cs.umd.edu

Abstract

We propose hinge-loss Markov random fields *(HL-MRFs), a powerful class of continuous-valued graphical models, for high-level computer vision tasks. HL-MRFs are characterized by log-concave density functions, and are able to perform efficient, exact inference. Their templated hinge-loss potential functions naturally encode soft-valued logical rules. Using the declarative modeling language* probabilistic soft logic*, one can easily define HL-MRFs via familiar constructs from first-order logic. We apply HL-MRFs to the task of activity detection, using principles of collective classification. Our model is simple, intuitive and interpretable. We evaluate our model on two datasets and show that it achieves significant lift over the low-level detectors.*

1. Introduction

In many computer vision tasks, it is useful to combine structured, high-level reasoning with low-level predictions. Collective reasoning at a high-level can take advantage of accurate low-level detectors, while improving the accuracy of predictions based on less accurate detectors. To fully leverage the power of high-level reasoning, we require a tool that is both powerful enough to model complex, structured problems and expressive enough to easily encode high-level ideas. In this work, we apply *hinge-loss Markov random fields* (HL-MRFs) [2, 3] to a high-level vision task. HL-MRFs are powerful, templated graphical models that admit efficient, exact inference over continuous variables. We demonstrate that, when combined with the modeling language *probabilistic soft logic* (PSL) [7, 17], HL-MRFs allow us to design high-level, structured models that improve the performance of low-level detectors.

We focus on the task of collective *activity detection* of humans in video scenes. Since human activities are often interactive or social in nature, collective reasoning over activities can provide more accurate detections than independent, local predictions. For instance, one can use aggregate predictions within the scene or frame to reason about the local actions of each actor. Further, collective models let us reason across video frames, to allow predictions in adjacent frames to inform each other, and thus implement the intuition that actions are temporally continuous.

We demonstrate the effectiveness of HL-MRFs and PSL on two group activity datasets. Using a simple, interpretable model, we are able to achieve significant lift in accuracy from low-level predictors. We thus show HL-MRFs to be a powerful, expressive tool for high-level computer vision.

1.1. Related Work

Motivated by the rich spatiotemporal structure of human activity, recent work in high-level computer vision has focused on modeling the complex interactions among observations explicitly, solving multiple vision problems *jointly*. These interactions could be between scenes and actions [21], objects and actions [13, 28], or actions performed by two or more people [8, 9, 18, 19]. They have been modeled using *context-free grammars* [25], AND-OR graphs [1, 14], probabilistic first-order logic [6, 22], and as network flow [10, 15, 16].

While most of these approaches require that person bounding boxes are pre-detected and pre-tracked to incorporate temporal cues [6, 8, 9, 13, 14, 18, 22, 25], recent work proposes solving activity recognition and tracking jointly. Khamis et al. [15] presented a network flow model to perform simultaneous action recognition and identity maintenance. They then augmented their model to jointly reason about scene types [16]. Similarly, Choi and Savarese [10] proposed a unified model to perform action recognition at the individual and group levels simultaneously with tracking. We build upon this work using a probabilistic relational approach.

PSL is one of many existing systems for probabilistic relational modeling, including *Markov logic networks* [24], *relational dependency networks* [23], and *relational Markov networks* [27], among others. One distinguishing feature of PSL is that its continuous representation of logical truth makes its underlying probabilistic model an HL-

MRF [3], which allows inference of the *most-probable explanation* (MPE) to be solved as a convex optimization. Our work benefits from recent advances on fast HL-MRF inference based on the *alternating direction method of multipliers* [2, 5], which significantly increases the scalability of HL-MRF inference over off-the-shelf convex optimization tools.

2. Hinge-loss Markov Random Fields

In this section we present *hinge-loss Markov random fields* (HL-MRFs), a general class of conditional, continuous-valued probabilistic models. HL-MRFs are log-linear probabilistic models whose features are hinge-loss functions of the variable states. Through constructions based on *soft logic* (explained in Section 3), hinge-loss potentials can be used to model generalizations of logical conjunction and implication, making these powerful models interpretable, flexible, and expressive.

HL-MRFs are parameterized by constrained hinge-loss energy functions.

Definition 1. *Let $\mathbf{Y} = (Y_1, \dots, Y_n)$ be a vector of n variables and $\mathbf{X} = (X_1, \dots, X_{n'})$ a vector of n' variables with joint domain $\mathbf{D} = [0,1]^{n+n'}$. Let $\phi = (\phi_1, \dots, \phi_m)$ be m continuous potentials of the form*

$$\phi_j(\mathbf{Y}, \mathbf{X}) = [\max\{\ell_j(\mathbf{Y}, \mathbf{X}), 0\}]^{p_j}$$

where ℓ_j is a linear function of $\mathbf{Y}$ and $\mathbf{X}$ and $p_j \in \{1,2\}$. Let $C = (C_1, \dots, C_r)$ be linear constraint functions associated with index sets denoting equality constraints $\mathcal{E}$ and inequality constraints $\mathcal{I}$, which define the feasible set

$$\tilde{\mathbf{D}} = \left\{ \mathbf{Y}, \mathbf{X} \in \mathbf{D} \middle| \begin{array}{l} C_k(\mathbf{Y}, \mathbf{X}) = 0, \forall k \in \mathcal{E} \\ C_k(\mathbf{Y}, \mathbf{X}) \geq 0, \forall k \in \mathcal{I} \end{array} \right\} .$$

For $\mathbf{Y}, \mathbf{X} \in \tilde{\mathbf{D}}$, given a vector of nonnegative free parameters, i.e., weights, $\lambda = (\lambda_1, \dots, \lambda_m)$, a constrained hinge-loss energy function *f_λ is defined as*

$$f_\lambda(\mathbf{Y}, \mathbf{X}) = \sum_{j=1}^{m} \lambda_j \phi_j(\mathbf{Y}, \mathbf{X}) .$$

Definition 2. *A* hinge-loss Markov random field *P over random variables $\mathbf{Y}$ and conditioned on random variables $\mathbf{X}$ is a probability density defined as follows: if $\mathbf{Y}, \mathbf{X} \notin \tilde{\mathbf{D}}$, then $P(\mathbf{Y}|\mathbf{X}) = 0$; if $\mathbf{Y}, \mathbf{X} \in \tilde{\mathbf{D}}$, then*

$$P(\mathbf{Y}|\mathbf{X}) = \frac{1}{Z(\lambda)} \exp\left[-f_\lambda(\mathbf{Y}, \mathbf{X})\right] , \qquad (1)$$

where $Z(\lambda) = \int_{\mathbf{Y}} \exp\left[-f_\lambda(\mathbf{Y}, \mathbf{X})\right]$.

The potential functions and weights can be grouped together into *templates*, which are used to define general classes of HL-MRFs that are parameterized by the structure of input data. Let $\mathcal{T} = (t_1, \dots, t_s)$ denote a vector of templates with associated weights $\Lambda = (\Lambda_1, \dots, \Lambda_s)$. We partition the potentials by their associated templates and let

$$\Phi_q(\mathbf{Y}, \mathbf{X}) = \sum_{j \in t_q} \phi_j(\mathbf{Y}, \mathbf{X})$$

for all $t_q \in \mathcal{T}$. In the *ground* HL-MRF, the weight of the j'th hinge-loss potential is set to the weight of the template from which it was derived, i.e., $\lambda_j = \Lambda_q$, for each $j \in t_q$.

MPE inference in HL-MRFs is equivalent to finding the feasible minimizer of the convex energy f_λ. Here, HL-MRFs have a distinct advantage over general discrete models, since minimizing f_λ is a convex optimization, rather than a combinatorial one.

Bach et al. [2] showed how to minimize f_λ using a consensus-optimization algorithm, based on the *alternating direction method of multipliers* (ADMM) [5]. Consensus-optimization works by creating local copies of the variables in each potential and constraint, constraining them to be equal to the original variables, and relaxing those equality constraints to make independent subproblems. By iteratively solving the subproblems and averaging the results, the algorithm reaches a consensus on the best values of the original variables, also called the *consensus variables*. This procedure is guaranteed to converge to the global minimizer of f_λ. (See [5] for more details on consensus optimization and ADMM.)

The inference algorithm has since been generalized and improved [3]. Experimental results suggest that the running time of this algorithm scales linearly with the size of the problem. On modern hardware, the algorithm can perform exact MPE inference with hundreds of thousands of variables in just a few seconds.

2.1. Weight Learning

To learn the parameters Λ of an HL-MRF given a set of training examples, we perform *maximum-likelihood estimation* (MLE), using the *voted perceptron* algorithm [11]. The partial derivative of the log of Equation 1 with respect to a parameter Λ_q is

$$\frac{\partial \log P(\mathbf{Y}|\mathbf{X})}{\partial \Lambda_q} = \mathbb{E}_\Lambda\left[\Phi_q(\mathbf{Y}, \mathbf{X})\right] - \Phi_q(\mathbf{Y}, \mathbf{X}) , \qquad (2)$$

where $\mathbb{E}_\Lambda$ is the expectation under the distribution defined by Λ. Note that the expectation in Equation 2 is intractable to compute. To circumvent this, we use a common approximation: the values of the potential functions at the most probable setting of $\mathbf{Y}$ with the current parameters [7]. The MPE approximation of the expectation is fast, due to the speed of the inference algorithm; however, there are no guarantees about its quality.

The voted perceptron algorithm optimizes Λ by taking steps of fixed length in the direction of the negative gradient, then averaging the points after all steps. To preserve the nonnegativity of the weights, any step that is outside the feasible region is projected back before continuing. For a smoother ascent, it is often helpful to divide the q-th component of the gradient by the number of groundings $|t_q|$ of the q'th template [20], which we do in our experiments.

3. Probabilistic Soft Logic

In this section, we review *probabilistic soft logic* (PSL) [7, 17], a declarative language for probabilistic reasoning. While PSL borrows the syntax of first-order logic, semantically, all variables take *soft truth values* in the interval $[0, 1]$, instead of only the extremes, 0 (FALSE) and 1 (TRUE). Continuous variables are useful both for modeling continuous domains as well as for expressing confidences in discrete predictions, which are desirable for the same reason that practitioners often prefer marginal probabilities to discrete MPE predictions. PSL provides a natural interface to design hinge-loss potential templates using familiar concepts from first-order logic.

A PSL program consists of a set of first-order logic rules with conjunctive bodies and disjunctive heads. Rules are constructed using the logical operators for conjunction ($\wedge$), negation ($\neg$) and implication ($\Rightarrow$). Rules are assigned weights, which can be learned from observed data. Consider the following rule for collective image segmentation.

$$0.8 : \textsc{Close}(P_1, P_2) \wedge \textsc{Label}(P_1, C) \Rightarrow \textsc{Label}(P_2, C)$$

In this example, P_1, P_2 and C are variables representing two pixels and a category; the predicate $\textsc{Close}(P_1, P_2)$ measures the degree to which P_1, P_2 are "close" in the image; $\textsc{Label}(P_1, C)$ indicates the degree to which P_1 belongs to class C, and similarly for $\textsc{Label}(P_2, C)$. This rule has weight 0.8.

PSL uses the *Lukasiewicz t-norm*, and its corresponding *co-norm*, to relax the logical operators for continuous variables. These relaxations are exact at the extremes, but provide a consistent mapping for values in between. For example, given variables X and Y, the relaxation of the conjunction $X \wedge Y$ would be $\max\{0, X + Y - 1\}$.

We say that a rule r is *satisfied* when the truth value of the head r_{HEAD} is at least as great as that of the body r_{BODY}. The rule's *distance from satisfaction* d_r measures the degree to which this condition is violated:

$$d_r = \max\{0, r_{\text{BODY}} - r_{\text{HEAD}}\}. \tag{3}$$

This corresponds to one minus the truth value of $r_{\text{BODY}} \Rightarrow r_{\text{HEAD}}$ when the variables are $\{0, 1\}$-valued. In the process known as *grounding*, each rule is instantiated for all possible substitutions of the variables as given by the data. For example, the above rule would be grounded for all pairs of pixels and categories.[1]

Notice that Equation 3 corresponds to a convex hinge function. In fact, each rule corresponds to a particular template $t \in \mathcal{T}$, and each grounded rule corresponds to a potential in the ground HL-MRF. If we let $\mathbf{X}_{i,j}$ denote the closeness of pixels p_i, p_j, and $Y_{i,c}$ denote the degree to which p_i has label c (likewise for p_j), then the example rule above would correspond to the potential function

$$\phi(\mathbf{Y}, \mathbf{X}) = [\max\{0, X_{i,j} + Y_{i,c} - Y_{j,c} - 1\}]^p,$$

where $p \in \{1, 2\}$ is the exponent parameter (see Definition 1). Thus, PSL, via HL-MRFs, defines a log-linear distribution over possible interpretations of the first-order rules.

Because it is backed by HL-MRFs, PSL has some additional features that are useful for modeling. The constraints in Definition 1 allow the encoding of functional modeling requirements, which can be used to enforce mutually exclusion constraints (i.e., that the soft-truth values should sum to one). Further, the exponent parameter p allows flexibility in the shape of the hinge, affecting the sharpness of the penalty for violating the logical implication. Setting p to 1 penalizes violation linearly with the amount the implication is unsatisfied, while setting p to 2 penalizes small violations much less. In effect, some linear potentials overrule others, while the influences of squared potentials are averaged together.

4. Collective Activity Detection

In this section, we apply HL-MRFs to the task of collective activity detection. We treat this as a high-level vision task, using the output of primitive, local models as input to a collective model for joint reasoning. We begin by describing the datasets and objective. We then describe our model. We conclude with a discussion of our experimental results.

4.1. Datasets

We use the collective activity dataset from [8] and its augmentation from [9] to evaluate our model. The first dataset contains 44 video sequences, each containing multiple actors performing activities in the set: *crossing*, *standing*, *queueing*, *walking*, and *talking*. The second dataset contains 63 sequences, with actions in: *crossing*, *standing*, *queueing*, *talking*, *dancing*, and *jogging*.[2] From each dataset, we use the bounding boxes (with position, width and height), pixel data, actions and identity annotations; we

[1] Though this could possibly lead to an explosion of groundings, PSL uses lazy activation to only create groundings for substitutions when the truth value of the body exceeds a certain margin.

[2] The *walking* action was removed from the augmented dataset by [9] because it was deemed ill-defined.

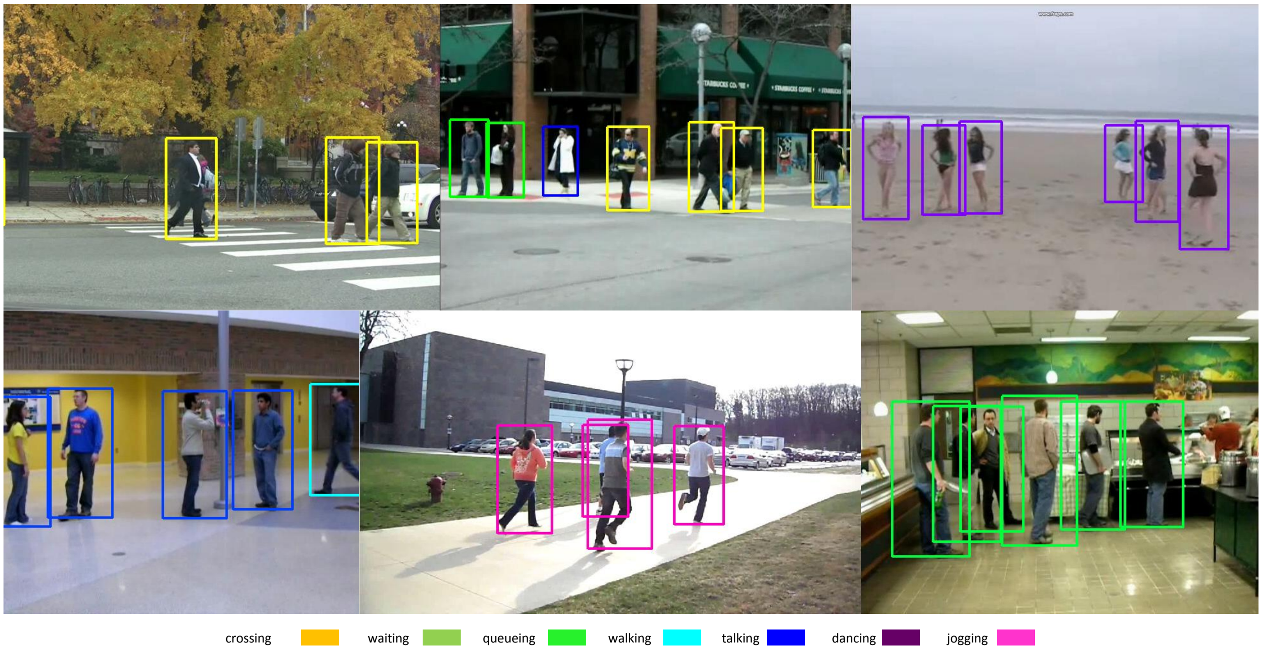

Figure 1. A few sample frames from the collective activity datasets. The original dataset and its augmentation include multiple actors in a natural setting performing specific actions. The colors of the bounding boxes in the figure specify the groundtruth action of the corresponding person.

do not use the 3-D trajectories. Activity detection in these datasets is challenging, since the scenes involve multiple actors in a natural setting; other action datasets, like KTH [26] or Weizmann [4], have a single person performing a specific action. In addition, there is considerable ambiguity in the actions being considered; for example, the actions *standing* and *queueing* are difficult to distinguish, even for a human. Figure 1 illustrates some sample frames from the two datasets.

Similar to [15, 16], we represent the detected human figures using *histogram of oriented gradients* (HOG) [12] features and *action context* (AC) descriptors [18]. The AC descriptor is a feature representation that combines the local beliefs about an actor's activities with those of actors in surrounding spatiotemporal neighborhoods. To create the AC descriptors, we use HOG features as the underlying feature representation; we then train a first-level SVM classifier on these features and combine the outputs per [18]. Finally, we train a second-stage SVM classifier on the AC descriptors to obtain the activity beliefs used in our high-level model. All classifiers are trained using a leave-one-out methodology, such that the predictions for the i'th sequence are obtained by training on all other sequences.

4.2. Model

Our primary objective is to enhance the low-level activity detectors with high-level, global reasoning. To do so, we augment the local features (described below) using relational information within and across adjacent frames. By modeling the relationships of bounding boxes, we can leverage certain intuitions about human activity. For instance, it is natural to assume that one's activity is temporally continuous; that is, it is not likely to change between points close in time. Further, there are certain activities that involve interaction with others, such as *talking* or *queueing*. Therefore, if we believe that one or more actors are talking, then actors nearby are also likely to be talking. Using PSL, modeling these intuitions is a simple matter of expressing them in first-order logic. We can then use HL-MRFs to reason *jointly* over these rules.

Our PSL model is given below.

$$\textsc{Local}(B,a) \Rightarrow \textsc{Doing}(B,a) \qquad \text{(R1)}$$

$$\textsc{Frame}(B,F) \wedge \textsc{FrameLabel}(F,a) \Rightarrow \textsc{Doing}(B,a) \qquad \text{(R2)}$$

$$\textsc{Close}(B_1,B_2) \wedge \textsc{Doing}(B_1,a) \Rightarrow \textsc{Doing}(B_2,a) \qquad \text{(R3)}$$

$$\textsc{Seq}(B_1,B_2) \wedge \textsc{Close}(B_1,B_2) \Rightarrow \textsc{Same}(B_1,B_2) \qquad \text{(R4)}$$

$$\textsc{Same}(B_1,B_2) \wedge \textsc{Doing}(B_1,a) \Rightarrow \textsc{Doing}(B_2,a) \qquad \text{(R5)}$$

Rule R1 corresponds to beliefs about local predictions (on either the HOG features or AC descriptors). R2 expresses the belief that if many actors in the current frame are doing a particular action, then perhaps everyone is doing that action. To implement this, we derive a FRAMELABEL predicate for each frame; this is computed by accumulating and normalizing the LOCAL activity beliefs for all actors in the frame. Similarly, R3 enforces our intuition about the effect of proximity on activity, where actors that are close[3] in the same frame are likely to perform the same action. This can be considered a fine-grained version of the second rule. R4 is used for identity maintenance and tracking. It essentially says that if two bounding boxes occur in adjacent frames and their positions have not changed significantly, then they are likely the same actor. We then reason, in R5, that if two bounding boxes (in adjacent frames) refer to the same actor, then they are likely to be doing the same activity. Note that rules involving lowercase a are defined for each action a, such that we can learn different weights for different actions. We define priors over the predicates SAME and DOING, which we omit for space. We also define (partial) functional constraints (not shown), such that the truth-values over all actions (respectively, over all adjacent bounding boxes), sum to (at most) one. We train the weights for these rules using 50 iterations of voted perceptron, with a step size of 0.1.

Note that we perform identity maintenance only to improve our activity predictions. During prediction, we do not observe the SAME predicate, so we have to predict it. We then use these predictions to inform the rules pertaining to activities.

4.3. Experiments

To illustrate the lift one can achieve on low-level predictors, we evaluate two versions of our model: the first uses activity beliefs from predictions on the HOG features; the second uses activity beliefs predicted on the AC descriptors. Essentially, this determines which low-level predictions are used in the predicates LOCAL and FRAMELABEL. We denote these models by HL-MRF + HOG and HL-MRF + ACD respectively. We compare these to the predictions made by the first-stage predictor (HOG) and the second-stage predictor (ACD).

The results of these experiments are listed in Table 1. We also provide recall matrices (row-normalized confusion matrices) for HL-MRF + ACD in Figure 2. For each dataset, we use leave-one-out cross-validation, where we train our model on all except one sequence, then evaluate our predictions on the hold-out sequence. We report cumulative accuracy and F1 to compensate for skew in the size and label distribution across sequences; this involves accumulating the confusion matrices across folds.

Our results illustrate that our models are able to achieve significant lift in accuracy and F1 over the low-level detectors. Specifically, we see that HL-MRF + HOG achieves a 12 to 20 point lift over the baseline HOG model, and HL-MRF + ACD obtains a 1.5 to 2.5 point lift over the ACD model.

Table 1. Results of experiments with the 5- and 6-activity datasets, using leave-one-out cross-validation. The first dataset contains 44 sequences; the second, 63 sequences. Scores are reported as the cumulative accuracy/F1, to account for size and label skew across folds.

	5 Activities		6 Activities	
Method	Acc.	F1	Acc.	F1
HOG	.474	.481	.596	.582
HL-MRF + HOG	**.598**	**.603**	**.793**	**.789**
ACD	.675	.678	.835	.835
HL-MRF + ACD	**.692**	**.693**	**.860**	**.860**

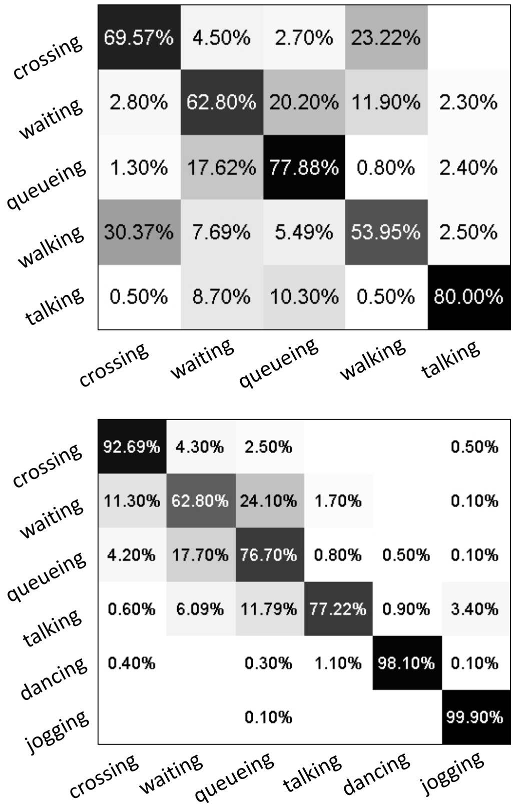

Figure 2. Recall matrices (i.e., row-normalized confusion matrices) for the 5- and 6-activity datasets, using the HL-MRF + ACD model.

5. Conclusion

We have shown that HL-MRFs are a powerful class of models for high-level computer vision tasks. When combined with PSL, designing probabilistic models is easy and intuitive. We applied these models to the task of collective activity detection, building on local, low-level detectors to create a global, relational model. Using simple, interpretable first-order logic rules, we were able to improve the accuracy of low-level detectors.

[3] To measure closeness, we use an RBF kernel.

Acknowledgements

This work was partially supported by NSF grants IIS1218488 and CCF0937094, NSF CAREER grant 0746930, and MURI from the Office of Naval Research under grant N00014-10-1-0934.

References

[1] M. R. Amer, D. Xie, M. Zhao, S. Todorovic, and S. C. Zhu, "Cost-sensitive top-down/bottom-up inference for multiscale activity recognition," in *European Conference on Computer Vision*, 2012. 1

[2] S. Bach, M. Broecheler, L. Getoor, and D. O'Leary, "Scaling MPE inference for constrained continuous markov random fields with consensus optimization," in *Neural Information Processing Systems*, 2012. 1, 2

[3] S. H. Bach, B. Huang, B. London, and L. Getoor, "Hinge-loss markov random fields: Convex inference for structured prediction," in *Uncertainty in Artificial Intelligence*, 2013. 1, 2

[4] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *ICCV*, 2005. 4

[5] S. Boyd, N. Parikh, E. Chu, B. Peleato, and J. Eckstein, "Distributed optimization and statistical learning via the alternating direction method of multipliers," *Foundations and Trends in Machine Learning*, vol. 3, no. 1, 2011. 2

[6] W. Brendel, S. Todorovic, and A. Fern, "Probabilistic event logic for interval-based event recognition," in *CVPR*, 2011. 1

[7] M. Broecheler, L. Mihalkova, and L. Getoor, "Probabilistic similarity logic," in *Uncertainty in Artificial Intelligence*, 2010. 1, 2, 3

[8] W. Choi, K. Shahid, and S. Savarese, "What are they doing?: Collective activity classification using spatio-temporal relationship among people," in *VS*, 2009. 1, 3

[9] W. Choi, K. Shahid, and S. Savarese, "Learning context for collective activity recognition," in *CVPR*, 2011. 1, 3

[10] W. Choi and S. Savarese, "A unified framework for multi-target tracking and collective activity recognition," in *European Conference on Computer Vision*, 2012. 1

[11] M. Collins, "Discriminative training methods for hidden Markov models: theory and experiments with perceptron algorithms," in *Empirical Methods in Natural Language Processing*, 2002. 2

[12] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005. 4

[13] A. Gupta and L. S. Davis, "Objects in action: An approach for combining action understanding and object perception," in *CVPR*, 2007. 1

[14] A. Gupta, P. Srinivasan, J. Shi, and L. S. Davis, "Understanding videos, constructing plots learning a visually grounded storyline model from annotated videos," in *CVPR*, 2009. 1

[15] S. Khamis, V. I. Morariu, and L. S. Davis, "A flow model for joint action recognition and identity maintenance," in *CVPR*, 2012. 1, 4

[16] S. Khamis, V. I. Morariu, and L. S. Davis, "Combining per-frame and per-track cues for multi-person action recognition," in *European Conference on Computer Vision*, 2012. 1, 4

[17] A. Kimmig, S. Bach, M. Broecheler, B. Huang, and L. Getoor, "A short introduction to probabilistic soft logic," in *NIPS Workshop on Probabilistic Programming: Foundations and Applications*, 2012. 1, 3

[18] T. Lan, Y. Wang, G. Mori, and S. N. Robinovitch, "Retrieving actions in group contexts," in *SGA*, 2010. 1, 4

[19] T. Lan, Y. Wang, W. Yang, and G. Mori, "Beyond actions: Discriminative models for contextual group activities," in *NIPS*, 2010. 1

[20] D. Lowd and P. Domingos, "Efficient weight learning for Markov logic networks," in *Principles and Practice of Knowledge Discovery in Databases*, 2007. 3

[21] M. Marszalek, I. Laptev, and C. Schmid, "Actions in context," in *CVPR*, 2009. 1

[22] V. I. Morariu and L. S. Davis, "Multi-agent event recognition in structured scenarios," in *CVPR*, 2011. 1

[23] J. Neville and D. Jensen, "Relational dependency networks," *Journal of Machine Learning Research*, vol. 8, pp. 653–692, 2007. 1

[24] M. Richardson and P. Domingos, "Markov logic networks," *Machine Learning*, vol. 62, no. 1-2, pp. 107–136, 2006. 1

[25] M. S. Ryoo and J. K. Aggarwal, "Stochastic representation and recognition of high-level group activities," *IJCV*, vol. 93, no. 2, pp. 183–200, 2010. 1

[26] C. Schuldt, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *ICPR*, 2004. 4

[27] B. Taskar, M. Wong, P. Abbeel, and D. Koller, "Link prediction in relational data," in *Neural Information Processing Systems*, 2003. 1

[28] B. Yao and L. Fei-Fei, "Modeling mutual context of object and human pose in human-object interaction activities," *CVPR*, 2010. 1