Learning Ensembles of Potential Functions for Structured Prediction with Latent Variables

Hossein Hajimirsadeghi and Greg Mori
School of Computing Science, Simon Fraser University, Canada
hosseinh@sfu.ca, mori@cs.sfu.ca

Abstract

Many visual recognition tasks involve modeling variables which are structurally related. Hidden conditional random fields (HCRFs) are a powerful class of models for encoding structure in weakly supervised training examples. This paper presents HCRF-Boost, a novel and general framework for learning HCRFs in functional space. An algorithm is proposed to learn the potential functions of an HCRF as a combination of abstract nonlinear feature functions, expressed by regression models. Consequently, the resulting latent structured model is not restricted to traditional log-linear potential functions or any explicit parameterization. Further, functional optimization helps to avoid direct interactions with the possibly large parameter space of nonlinear models and improves efficiency. As a result, a complex and flexible ensemble method is achieved for structured prediction which can be successfully used in a variety of applications. We validate the effectiveness of this method on tasks such as group activity recognition, human action recognition, and multi-instance learning of video events.

1. Introduction

Challenging structured vision problems necessitate the use of high-capacity models. Examples include problems such as modeling group activities or temporal dynamics in human action recognition and internet video analysis. Recently, visual recognition has made great strides using deep models. Deep learning has been successfully applied to image classification [18, 29] and object detection [12]. This success arises from large-scale training of highly non-linear functions which can induce complex models and learn powerful abstract feature representations. However, learning non-linear functions for structured vision problems remains an open challenge. In this paper, we present a general method for learning non-linear representations for structured models.

Our method works within a graphical model framework, building an HCRF to model structure, as depicted in Fig. 1.

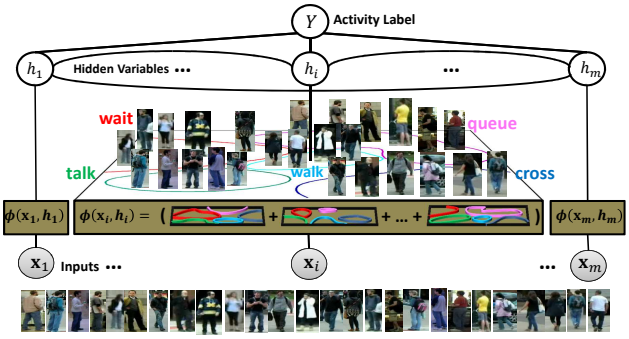


Figure 1. The proposed method learns non-linear potential functions in a latent structured model. An example model for group activity is shown. Potential functions relate input image regions to variables such as body pose or action/activity. Each potential function is learned as a combination of non-linear models leading to a high-capacity model. The colored ribbon-like lines visualize the decision boundaries obtained by nonlinear potential functions.

Recent efforts in this vein [32, 3, 28] have attempted to design unified deep structured models by equipping Markov random fields (MRFs) with the representational power of convolutional neural networks (CNNs). These methods jointly train an MRF and a CNN by maximizing likelihood via back-propagation and stochastic gradient descent. However, all these methods are defined for fully observed output variables and cannot incorporate or infer dependencies on unlabeled variables in the case of weak supervision. Full annotation of all output variables in MRFs is very costly for many visual recognition tasks, and hence many variables remain latent, unobserved, in training.

The standard learning algorithms for latent structured models (e.g. latent SVM [9] or HCRF [26]) are restricted to simple log-linear models, where the potential functions are parameterized by linear combination of the input features. Thus, they lack the non-linearity and feature abstraction power of deep models. In this work, we alleviate this problem by proposing a general framework to learn latent structured models with arbitrary potential functions in func-

tional space.

We propose an algorithm based on functional gradient ascent (i.e., gradient boosting). By using this functional approach, training a latent structured model is decoupled from explicit representation of feature interactions in the potentially large parameter space of the potential functions. This provides scalability and improves efficiency [7]. This decoupling helps to define potential functions as a combination of new abstract features encoded by nonlinear regression models such as regression trees, kernel support vector machines, or deep neural networks. As a result, a highly complex model can be achieved with an efficient learning algorithm. In addition, because of the *ensemble effect* of combining numerous base models, the proposed method is less prone to overfitting.

2. Related Work

In this section, we review related work within learning algorithms for structured prediction and their use in computer vision.

Learning algorithms for structured prediction: Conditional random fields (CRFs) are among the primary tools for structured visual recognition. Nonlinear variants of CRFs include kernel conditional random fields [19], and CRFs with deep neural network features [8]. Dietterich et al. [7] and Chen et al. [4] proposed a boosting framework to train CRFs with abstract features represented by regression trees. Jancsary et al. [16] introduced regression tree fields, a Gaussian CRF model parameterized by regression trees. Tompson et al. [32], Chen et al. [3], Schwing and Urtasun [28] proposed methods to combine convolutional neural networks with CRF-based graphical models for deep structured prediction. Deng et al. [6] proposed a deep neural network with layers which mimic message-passing steps in probabilistic graphical models.

Hidden conditional random fields [26] learn CRFs with latent variables by maximizing the likelihood function marginalized over the hidden variables via gradient ascent. Max-margin variants of HCRF (a.k.a. latent SVM) [9, 44, 38] use alternating minimization strategies. Schwing et al. [27] proposed a general structured loss minimization framework for structured prediction with latent variables. All these algorithms are used for learning log-linear models, which limits their ability to model complex prediction tasks.

Nonlinear extensions of these algorithms have been proposed based on predefined kernels, e.g. kernelized latent SVM [41], kernels on CRFs [15], or non-linear feature encoding techniques [34]. However, the kernelized latent SVM methods have high computational complexity and lack efficient inference algorithms, resorting to enumeration over (single) latent variables. The CRF kernel method uses log-linear models trained similar to the standard HCRF [26].

In contrast, our work presents a general framework for learning latent structured models, which trains HCRFs with arbitrary potential functions represented by an ensemble of nonlinear base models. Thus, it can represent richer dependencies between the variables, be integrated with a variety of base models, and provide efficient learning and inference algorithms; empirically we show these can deliver superior recognition performance.

Structured prediction for group activity: Structured prediction has been extensively used in a variety of computer vision applications. A series of recent papers has focused on the problem of group activity recognition, inferring an activity that is performed by a set of people in a scene. Choi et al. [5], Lan et al. [21], and Khamis et al. [17] devised models for spatial and temporal relations between the individuals involved in a putative interaction. Lan et al. [21] proposed latent CRF models with optimized graph structures for joint action-activity recognition. Amer et al. [1] proposed a hierarchical random field to jointly model temporal and frame-wise relations of video features describing an activity in a hierarchy of mid-level video representations.

Individual human action recognition: A variety of feature descriptors has been designed to extract discriminative spatio-temporal information from depth sequences. For example, Yang et al. [43] proposed new HOG descriptors built on depth motion maps. Wang et al. [37] trained an actionlet ensemble model based on novel local skeleton features to represent and recognize human actions. Xia and Aggarwal [39] introduced depth cuboid similarity features to make codewords for depth video recognition. Yang and Tian [42] proposed super normal vector (SNV) to describe a depth sequence with a codebook of polynormals obtained by clustering surface normals in the sequence. We perform empirical evaluation on action recognition from depth data, showing the efficacy of our learning approach. Similar to some previous works in action/gesture recognition, we use graphical models to capture the temporal dynamics and motion patterns of action [26, 22, 13].

Unconstrained internet video analysis: Structural models have been also successfully used for unconstrained internet video analysis. Methods to capture the temporal structure of high-level events need to be robust to the presence of irrelevant frames. Successful models include Tian et al. [31] and Niebles et al. [23], who extended latent variable models in the temporal domain. Vahdat et al. [33] composed a test video with a set of kernel matches to training videos. Tang et al. [30] effectively combined informative subsets of features extracted from videos to improve event detection. Pirsiavash and Ramanan [25] developed stochastic grammars for understanding structured events. Xu et al. [40] proposed a feature fusion method based on utilizing related exemplars for event detection. Lai et al. [20] applied multi-instance

learning to video event detection by representing a video as multi-granular temporal video segments.

3. Proposed Method: HCRF-Boost

We propose a general framework for learning non-linear latent structured models. A high-level overview of our proposed method is as follows. We need to learn potential functions for a structured model over inputs, latent variables, and outputs. These potential functions control compatibilities between various settings of the variables – e.g. the relationships between image observations and their class labels. In order to model challenging problems, complex non-linear relationships between these variables are needed.

Figure 2 shows our proposed HCRF-Boost model. The potential functions are defined as a combination of multiple nonlinear functions, obtained stage by stage. To find these functions we use functional gradient ascent (i.e. gradient boosting). Gradient boosting is the functional analog of the standard gradient ascent. At each step, a functional gradient is found by taking the derivatives of the objective function (likelihood function in our case) directly w.r.t the potential functions (instead of the parameters). So, at each step a new function g_t is derived, where the potential function should move in that functional direction. In this paper, we show how to take these derivatives efficiently and approximate the functional gradients with nonlinear fitting functions. In the following sections the preliminaries and details of the proposed method are explained. A summary of the resulting algorithm is given in Alg. 1.

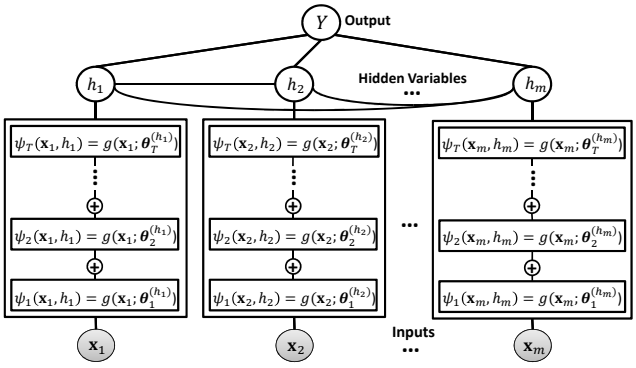


Figure 2. Latent structured prediction with our proposed HCRF-Boost model. Note that there exist potentials on all edges. But, the potentials between the hidden and output variables are not shown in this graph for clarity of illustration.

3.1. Preliminaries

Due to space limitations, we provide very brief summaries of gradient boosting [10] and HCRFs [26] below. Please see the corresponding references for more details.

Gradient Boosting: Gradient boosting learns a classifier $F(\mathbf{x}) = \sum_t \beta_t f_t(\mathbf{x})$ by optimizing an objective function $\mathcal{L}(y, F(\mathbf{x}))$ in a functional space by performing gradient ascent. The optimization is approximated by a greedy stage-wise optimization of the form

$$(\beta_t, f_t) = \arg\min_{\beta, f} \sum_{n=1}^{N} \mathcal{L}(y^n, F_{t-1}(\mathbf{x}^n) + \beta\, f(\mathbf{x}^n)). \quad (1)$$

using a training set $\{(\mathbf{x}^1, y^1), \cdots, (\mathbf{x}^N, y^N)\}$. To solve this problem, first the so-called pseudo-residuals are computed for each training instance as

$$\widehat{f}(\mathbf{x}^n) = \frac{\partial \mathcal{L}(y^n, F(\mathbf{x}^n))}{\partial F(\mathbf{x}^n)}|_{F(\mathbf{x})=F_{t-1}(\mathbf{x})} \quad (2)$$

After computing the pseudo-residuals, a new base classifier $f_t(\mathbf{x})$ is trained by fitting a regression model to the training set $\{(\mathbf{x}^n, \widehat{f}(\mathbf{x}^n))\}_n$, i.e., $f_t : \mathbf{x}^n \rightarrow \widehat{f}(\mathbf{x}^n)$. Given this function fixed, the multiplier β_t is found simply by doing a line search. It has been shown that since a whole model is added at each iteration of gradient boosting, a big step can be taken to maximize the objective function [7].

Hidden Conditional Random Fields: A hidden conditional random field (HCRF) is defined on a 3-tuple ($\mathbf{X} \in \mathcal{X}, \mathbf{h} \subset \mathcal{H}, Y \in \mathcal{Y}$), where $\mathbf{h}$ is the set of latent variables, which are not observed in the training data. Given this, the posterior probability distribution is obtained by

$$P(Y|\mathbf{X}) = \sum_{\mathbf{h}} P(Y, \mathbf{h}|\mathbf{X}) = \frac{\sum_{\mathbf{h}} \exp\left(F(\mathbf{X}, Y, \mathbf{h})\right)}{\sum_{Y', \mathbf{h}} \exp\left(F(\mathbf{X}, Y', \mathbf{h})\right)}, \quad (3)$$

where the whole graph potential function factorizes as

$$F(\mathbf{X}, Y, \mathbf{h}) = \sum_i \phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i). \quad (4)$$

In the standard HCRF model proposed by [26], the potential functions are linearly parameterized as

$$\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \boldsymbol{\gamma_i}(\mathbf{X}_i, Y_i, \mathbf{h}_i)\, \boldsymbol{\theta}_i. \quad (5)$$

and parameters are learned using maximum a posteriori estimation.

In this paper, we alleviate the limitation of parameterizing the HCRFs and learn the potential functions in a functional space, using a boosting approach. As a result, highly non-linear and powerful models can be achieved.

3.2. HCRF-Boost: Gradient Boosting of HCRFs

In this work, we use gradient boosting for training HCRF models. For this purpose, we maximize the likelihood function in (3) directly with respect to the clique potential functions. Consequently, each potential function is written as the combination of a number of *base potential functions*:

$$\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \sum_t \beta_t\, \psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i), \quad (6)$$

where each base potential function is estimated in a stage-wise manner by taking the derivatives of the log likelihood function w.r.t. the potential functions (given the current model estimation):

$$\widehat{\psi}_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \frac{\partial \log P(Y|\mathbf{X})}{\partial \phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i)}|_{f=f_{t-1}}. \quad (7)$$

We call this the pseudo-residual potential function. By plugging into the likelihood function of (3) and using the relations in [7], we get the following functional gradients at a given point $(\mathbf{X}^n, Y^n)$:

$$\begin{aligned}\widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i) = &\frac{\partial \log \sum_{\mathbf{h}} \exp\left(f(\mathbf{X}^n, Y^n, \mathbf{h})\right)}{\partial \phi_i(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)} \\ &- \frac{\partial \log \sum_{Y',\mathbf{h}} \exp\left(f(\mathbf{X}^n, Y', \mathbf{h})\right)}{\partial \phi_i(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)} \\ =&P(\mathbf{h}_i|\mathbf{X}^n, Y^n)\mathbb{1}(Y_i = Y_i^n) - P(\mathbf{h}_i, Y_i|\mathbf{X}^n) \\ &\forall\, i, Y_i, \mathbf{h}_i.\end{aligned} \quad (8)$$

Given the finite training set $\mathcal{D}^{tr} = \{(\mathbf{X}^n, Y^n)\}_{n=1}^N$ these are point-wise functional gradients, which are only defined at the training data points [10]. However, they provide the functional gradient training examples $\mathcal{D}_{i,t}^{(Y_i,\mathbf{h}_i)} = \left\{\left((\mathbf{X}^n, Y^n), \widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)\right)\right\}_n$, which can be fitted by a regression model in order to make smooth approximate pseudo-residual potential functions:

$$\begin{aligned}&\psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \\ &\arg\min_{\psi_i} \sum_n \left(\psi_i(\mathbf{X}_i^n, Y_i, \mathbf{h}_i) - \widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)\right)^2 \\ &\forall\, i, Y_i, \mathbf{h}_i.\end{aligned} \quad (9)$$

This fitting is done by learning the parameters of a regression model for each possible value of the output and hidden variables, i.e.,

$$\begin{aligned}&\psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i) = g(\mathbf{X}_i; \boldsymbol{\theta}_{i,t}^{(Y_i,\mathbf{h}_i)}), \\ &\boldsymbol{\theta}_{i,t}^{(Y_i,\mathbf{h}_i)} = \arg\min_{\boldsymbol{\theta}} \sum_n \left(g(\mathbf{X}_i^n; \boldsymbol{\theta}) - \widehat{\psi}_{i,t}(\mathbf{X}_i^n, Y_i, \mathbf{h}_i)\right)^2 \\ &\forall\, i, Y_i, \mathbf{h}_i.\end{aligned} \quad (10)$$

Hence, in the most general case, the number of trained models can grow exponentially with the number of variables in the largest clique. However, in practice, where common HCRF models are used, this procedure is reduced to training a few models (see next section). Finally, given the resulting functions, the potential function at the current iteration is updated as

$$\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) \leftarrow \phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) + \beta_t\, \psi_{i,t}(\mathbf{X}_i, Y_i, \mathbf{h}_i), \quad (11)$$

where the the step-length parameter β_t can be found by optimizing the likelihood function with a simple line search[1].

Algorithm 1 HCRF-Boost Algorithm

1: **Input:** Training data $\{(\mathbf{X}^n, Y^n)\}_{n=1}^N$.
2: Initialize the potential functions $\phi_i(\mathbf{X}_i, Y_i, \mathbf{h}_i) = \mathbf{0}$.
3: **repeat**
4: **for** each potential function ϕ_i **do**
5: Compute the pseudo-residual potentials $\widehat{\psi}_{i,t}$ according to (8) for all training examples.
6: Train new base potential functions $\psi_{i,t}$ according to (10) by fitting the input training examples to the pseudo-residual potentials.
7: Update the potential function: $\phi_i \leftarrow \phi_i + \beta_t\, \psi_{i,t}$.
8: **end for**
9: **until** converged or maximum number of iterations

3.3. HCRF-Boost for Unary and Pairwise Potentials

In the previous section, we described the HCRF-Boost algorithm for general HCRF models. In this section, a more detailed explanation of the algorithm is provided for HCRF models with unary and pairwise potentials, which are commonly used in visual recognition [26].

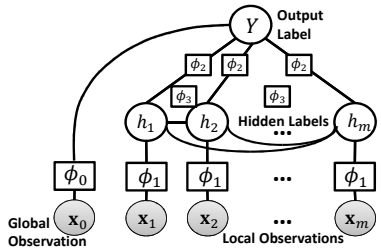


Figure 3. A hidden conditional random field with unary and pairwise potential functions.

A graphical representation of this model is shown in Figure 3. This graph is composed of the input observations $\mathbf{X} = \{\mathbf{x}_0, \mathbf{x}_1, \cdots, \mathbf{x}_m\}$, the output label Y, and the hidden labels $\mathbf{h} = \{h_1, \cdots, h_m\}$. The input observations are feature descriptors extracted from an image or video, where $\mathbf{x}_0$ is a global feature descriptor which represents the whole input, while $\mathbf{x}_i\,(i \neq 0)$ are local observations. Each local observation $\mathbf{x}_i$ is connected to its hidden label h_i. The connections between the hidden labels is represented by a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, where the edges $(i, j) \in \mathcal{E}$ denote the links between the hidden labels h_i and h_j. Finally, all hidden labels are linked to the output label Y. The goal is to predict the output label Y, given the input observations $\mathbf{X}$

[1] However, there is both theoretical and empirical evidence that this parameter can be safely set to a small constant value (e.g., 0.1) [2]. In all our experiments, we follow this rule.

and the structural constraints of the induced graph, by modeling the posterior probability $P(Y|\mathbf{X})$.

Given this model, the whole graph potential function takes the following form:

$$f(\mathbf{X}, Y, \mathbf{h}) = \phi_0(\mathbf{x}_0, Y) + \sum_{i=1}^{m} \phi_1(\mathbf{x}_i, h_i) + \sum_{i=1}^{m} \phi_2(Y, h_i) + \sum_{(i,j)\in\mathcal{E}} \phi_3(Y, h_i, h_j). \tag{12}$$

The learning process is to find the potential functions $\phi_0, \phi_1, \phi_2, \phi_3$ which maximize the likelihood function, by taking the functional gradients. Following the formula derived in (6), the pseudo-residuals of each potential function for a given data point $(\mathbf{X}^n, Y^n)$ at iteration t are obtained by[2]:

$$\widehat{\psi}_{0,t}(\mathbf{x}_0^n, Y) = \mathbb{1}(Y = Y^n) - P(Y|\mathbf{X}^n) \tag{13}$$

$$\widehat{\psi}_{1,t}(\mathbf{x}_i^n, h_i) = P(h_i|\mathbf{X}^n, Y^n) - P(h_i|\mathbf{X}^n) \tag{14}$$

$$\widehat{\psi}_{2,t}(Y, h_i) = P(h_i|\mathbf{X}^n, Y^n)\mathbb{1}(Y = Y^n) - P(h_i, Y|\mathbf{X}^n) \tag{15}$$

$$\widehat{\psi}_{3,t}(Y, h_i, h_j) = P(h_i, h_j|\mathbf{X}^n, Y^n)\mathbb{1}(Y = Y^n) - P(h_i, h_j, Y|\mathbf{X}^n) \tag{16}$$

$$\forall\, i \in \mathcal{V}, (i,j) \in \mathcal{E}, Y \in \mathcal{Y}, h_i \in \mathcal{H}.$$

Note that all these probabilities are the marginal probabilities which can be found by sum-product inference of the CRFs. For the popular CRF models that we use in this paper, such as tree-structured graphs or cardinality models, these marginals can be inferred exactly in linear or linearithmic time.

Next, by solving the fitting problem of (10), it can be shown that the smooth approximate functions are found as

$$\psi_{0,t}(\mathbf{x}_0, Y = a) = g(\mathbf{x}_0, \boldsymbol{\theta}_{0,t}^{(a)}) : \{\mathbf{x}_0^n \rightarrow \widehat{\psi}_{0,t}(\mathbf{x}_0^n, a)\}_{\mathcal{D}^{tr}} \quad \forall\, a \in \mathcal{Y} \tag{17}$$

$$\psi_{1,t}(\mathbf{x}_i, h_i = b) = g(\mathbf{x}_i, \boldsymbol{\theta}_{1,t}^{(b)}) : \{\mathbf{x}_i^n \rightarrow \widehat{\psi}_{1,t}(\mathbf{x}_i^n, b)\}_{\mathcal{D}^{tr},\mathcal{V}} \quad \forall\, b \in \mathcal{H} \tag{18}$$

$$\psi_{2,t}(Y = a, h_i = b) = \texttt{mean}\,\{\widehat{\psi}_{2,t}(a, b)\}_{\mathcal{D}^{tr},\mathcal{V}} \quad \forall\, a \in \mathcal{Y}, b \in \mathcal{H} \tag{19}$$

$$\psi_{3,t}(Y = a, h_i = b, h_j = c) = \texttt{mean}\,\{\widehat{\psi}_{3,t}(a, b, c)\}_{\mathcal{D}^{tr},\mathcal{E}} \quad \forall\, a \in \mathcal{Y}, b \in \mathcal{H}, c \in \mathcal{H}. \tag{20}$$

[2]Although the behind-the-scenes steps to derive the functional gradients are non-trivial, the results are intuitive. For example, (13) says that if Y is observed in the training data (i.e., $Y = Y^n$), $P(Y|\mathbf{X}^n)$ should be equal to 1 to make the subgradients zero and maximize the likelihood. Likewise, (14) says that the probability of the latent variables, with and without Y being observed, should become equal. In fact, these functional gradients are representing the errors but on a probability scale.

The first set of functions in (17) and (18) are trained by a regression model. So, only $|\mathcal{Y}| + |\mathcal{H}|$ functions should be trained. The next functions in (19) and (20) are simply obtained by taking the mean over all training examples. See the supplementary material for the detailed analysis of the computational complexity of the whole method.

3.4. Discussion

The fitting in (17) and (18) can be performed by training any regression model such as regression trees, kernel support vector machines, or even deep neural networks[3]. In practice training a support vector regression (SVR) model is faster than trees (especially for large feature vectors). Thus, in all our experiments we used SVR models. However, note that tree models can help for feature selection as well.

Further, for all the visual recognition tasks in Section 4, we use task-specific hand-crafted features. But, by using convolutional neural networks (CNNs) for model fitting, deep features can be also learned. In fact, employing CNNs with our method leads to an extension of the recent algorithms for learning deep structured models [28, 3]. These algorithms maximize the likelihood function $P(Y|\mathbf{X}, \mathbf{w}) = \frac{\exp(f(\mathbf{X},Y,\mathbf{w}))}{Z(\mathbf{X},\mathbf{w})}$ w.r.t. the parameters $\mathbf{w}$ via gradient ascent and backpropagation, where $f(\mathbf{X}, Y, \mathbf{w})$ is a CNN parameterized by $\mathbf{w}$. However, HCRF-Boost with CNNs extends these algorithms by (1) incorporating the structured hidden variables and (2) learning via functional gradient ascent (i.e. gradient boosting).

In our implementation, we used *stochastic gradient boosting* [11]. In this variation of gradient boosting, at each step, a random subset of training data is selected for computing the pseudo-residuals and fitting the base models. As a result, gradient boosting is combined with bagging (similar to random forest). Incorporating this randomization is advantageous for both improving the accuracy and speeding up the algorithm [11]. In all the experiments we subsampled 90% of data (without replacement) at each iteration.

4. Experiments

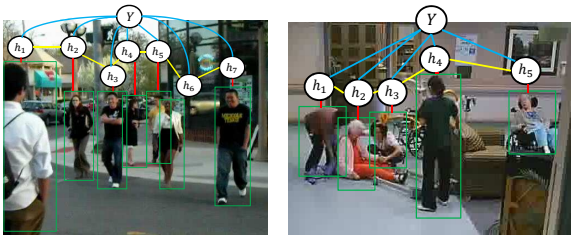
We provide empirical results on three different tasks: group activity recognition, human action recognition, and video event detection.

4.1. Spatial Structured Models: Group Activity Recognition

In this section, our proposed HCRF-Boost algorithm is used to train HCRFs which model spatial relations between individuals doing actions in a scene to recognize high-level group activities. Hence, the individual actions provide the

[3]Since the computed pseudo-residuals can be very small values, it is recommended to scale them to the range [0, 1] before doing regression.

context to infer the whole group activity. We run experiments on two datasets: collective activity dataset [5] and nursing home dataset [21]. Example HCRF models for this task are shown in Figure 4. This model is composed of nodes representing the people, actions, and the group activity. The hidden nodes are the individual actions which are linked to each other with a tree-structured graph, obtained by running maximum spanning tree.



(a) (b)

Figure 4. Group activity recognition with spatial structured models. (a) An example HCRF model from collective activity dataset. (b) An example HCRF model from nursing home dataset.

4.1.1 Collective Activity Dataset

The Collective Activity Dataset [5] comprises 44 videos (about 2500 video frames) of *crossing*, *waiting*, *queuing*, *walking*, and *talking*. Our goal is to classify the collective activity in each frame. Each person is represented by the *action context* feature descriptor proposed in [21]. We follow the same experimental settings as used in [21], i.e., the same 1/3 of the video clips were selected for test and the rest for training. As the latent models, we use the HCRF shown in Figure 4(a) with 5 hidden labels. The result of our method is shown in Table 1 and compared with the following methods[4]: (1) SVM on global bag-of-words, (2) latent SVM method in [21], and (3) HCRF (our own baseline). We also visualize some examples of recognition with our method in Figure 5.

Table 1. Comparison of classification accuracies of different algorithms on collective activity dataset. Both multi-class accuracy (MCA) and mean per-class accuracy (MPCA) are shown because of class size imbalance.

Method	MCA	MPCA
Global bag-of-words with SVM [21]	70.9	68.6
Latent SVM with optimized graph [21]	79.7	78.4
HCRF	76.2	75.2
HCRF-Boost (our proposed method)	**82.5**	**79.4**

[4]These methods follow the standard multiclass classification evaluation protocol in [5, 21], which is different from the binary classification in [1].

4.1.2 Nursing Home Dataset

In this section, we evaluate our method for activity recognition in a nursing home. The dataset we use [21] images scenes in which the individuals might be performing any of five actions: walking, standing, sitting, bending, or falling. However, the goal is to detect the whole scene activity, i.e., if any person is falling or not.

The dataset has 22 video clips (12 clips for training and 8 clips for test) with 2990 annotated frames, where about one third of them are assigned the "fall" activity label. We use the same feature descriptor as used in [21]. In short, this feature vector is obtained by concatenating the score of SVM classifiers trained for recognizing each of the five actions on the training dataset. Similar to the previous section, we use the HCRF model shown in Figure 4(b) with five hidden labels. The results in terms of classification accuracy and average precision are shown in Table 2. Again, we compare our method with a global bag-of-words model, latent SVM, and standard HCRF algorithm.

Table 2. Comparison of different algorithms on the nursing home dataset in terms of average precision (AP), mean per-class accuracy (MPCA), and multi-class accuracy (MCA). Note that because of the significant class size imbalance between the two classes, MCA is not an informative metric in this task

Method	AP	MPCA	MCA
Global bag-of-words [21]	43.3	52.4	48.0
Latent SVM [21]	48.8	67.4	71.5
HCRF	44.4	66.3	75.2
HCRF-Boost (ours)	**49.6**	**73.0**	**75.4**

4.2. Temporal Structured Models: Human Action Recognition

In this section, we apply our method for human action recognition with chain-structured HCRFs, capturing the temporal dynamics of the action. A graphical model of this task is illustrated in Figure 6. This HCRF consists of the input nodes, representing temporal segments of a depth sequence, connected to the hidden-state nodes. There is also a root potential function to globally model the interaction between the whole action sequence and the action label.

We evaluate the proposed model on the MSRAction3D dataset [22]. This dataset has 567 depth map sequences of 20 different actions performed by 10 subjects. The actions are movements common in gaming such as "hand catch", "forward punch", "draw tick", "tennis swing". As the features, we use the super normal vector (SNV) descriptors [42]. But, instead of the raw SNV features, we convert them into SVM scores and make a discriminative feature descriptor, as in Section 4.1.2.

The experiments were conducted by dividing each depth sequence into eight equal temporal segments and using the

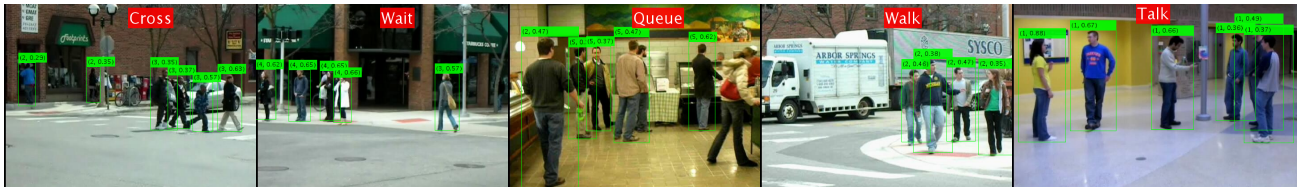


Figure 5. Examples of recognition with the proposed HCRF-Boost method. Each figure is annotated by the predicted collective activity. Also each individual is annotated by a tuple, indicating the inferred hidden label and its probability. Since the hidden labels are not observed during training, they have been represented symbolically by 1, 2, 3, 4, 5. However, interestingly, they have been learned to semantically categorize the individual actions (i.e., 1: talk; 2: walk; 3: cross; 4: wait; 5:queue). For example, in the first figure from left, four people are crossing the street while the two others are walking in the sidewalk. In the second figure, four people are waiting and one is crossing. In the third figure, four people are queuing in the line and one person is walking to join the lineup. In the fourth and fifth figures, all the individuals are walking and talking, respectively.

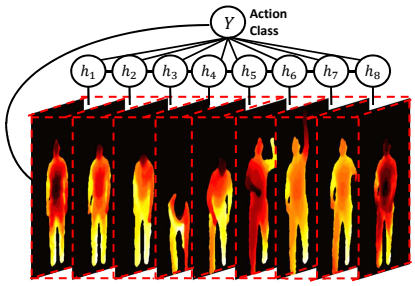


Figure 6. The HCRF model for human action recognition from a depth sequence.

HCRF model of Figure 6 with 5 hidden states for each segment. To have a fair comparison we followed the same experimental protocol as [42, 37]. The results are shown in Table 3 and compared with the state-of-the art methods for depth-based action recognition. Note that the global model[5] and HCRF algorithm are our own baselines.

Table 3. Comparison of classification accuracies of different algorithms on MSRAction3D dataset.

Method	Accuracy
Bag of 3D Points [22]	74.70%
Actionlet Ensemble [37]	88.20%
Depth Motion Maps [43]	88.73%
DSTIPv [39]	89.30%
Skeletal [35]	89.48%
Pose Set [36]	90.00%
Moving Pose [45]	91.70%
SNV [42]	93.09%
Our global model (using SNV)	92.73%
HCRF (using SNV)	91.64%
HCRF-Boost (using SNV)	**94.18%**

[5]Our global model is the same as the model proposed in [42] for SNV. However, we could not get the same accuracy (92.73 vs 93.09) with our duplication of their experiments.

4.3. Cardinality Models for Multi-Instance Learning: Multimedia Event Detection

Multiple instance learning (MIL) aims to recognize patterns from weakly supervised data. Contrary to standard supervised learning, where each training instance is labeled, in the MIL paradigm a *bag of instances* share a label, and the instance labels are hidden. Hajimirsadeghi et al. [14, 15] introduced HCRF models for MIL by incorporating cardinality-based potential functions. These cardinality potentials permit the modeling of the counts of inputs that contribute to an overall label.

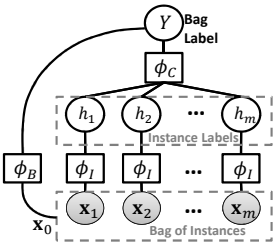


Figure 7. A graphical representation of the cardinality model. The instance labels are hidden variables.

A graphical representation of the cardinality model is shown in Figure 7. Each instance and its label are modeled by two nodes in a clique. The potential function of this clique (ϕ_I) specifies a classifier for an individual instance. There is also an optional clique potential between the global representation of the bag and the bag label (ϕ_B). Finally, a third clique potential (ϕ_C) contains all instance labels and the bag label. This clique is used to define what makes a bag positive or negative. Varying this clique potential will lead to different multi-instance assumptions. To this end, two different cardinality-based functions are defined, one for positive bags ($C^{(+1)}$) and one for negative bags ($C^{(-1)}$):

$$\phi_C(Y, \mathbf{h}) = C^{(Y)}(\sum_i h_i). \tag{21}$$

In general, $C^{(+1)}$ and $C^{(-1)}$ could be expressed by any cardinality function which can model MIL constraints. However, in our work we focus on the Normal cardinality model:

$$C^{(+1)}(c) = -(\frac{c}{m} - \mu)^2/2\sigma^2, \;\; C^{(-1)}(c) = -(\frac{c}{m})^2/2\sigma^2. \tag{22}$$

The parameter μ in this model controls the ratio of positive labeled instances in a positive bag.

In this work, we use our proposed HCRF-Boost to train these cardinality models. The experiments on popular MIL benchmark datasets and comparison with some state-of-the-art MIL methods are provided in the supplementary material. In this section, we evaluate our method for event detection on the challenging TRECVID MED11 dataset [24].

Recently, Lai et al. [20] proposed novel multi-instance methods (single–g $\propto$SVM and multi–g $\propto$SVM) for video event detection, by treating a video as a bag of temporal video segments of different granularity (single–g $\propto$SVM uses only single frames but multi–g $\propto$SVM uses both the single frames and video segments). Hajimirsadeghi et al. [15] followed a similar MIL approach to video event detection by embedding the cardinality models into a powerful kernel, "Cardinality Kernel." We evaluate the performance of our algorithm compared to these methods. In our framework, each video is treated as a bag of ten temporal video segments, where each segment is represented by pooling the features inside it. As the cardinality potential, we use the Normal model in (22) with $\mu = 1$ and $\sigma = 0.1$ to embed a soft and intuitive constraint on the number of positive instances: *the more relevant segments in a video, the higher the probability of the event occurring*.

Similar to the experiments in [20, 15], we use dense SIFT features quantized into bag-of-words vectors for each video segment[6]. The results are shown in Table (4). The HCRF method (used to train the cardinality model) performs poorly in this task because of using a linear feature representation. Our method outperforms multi–g $\propto$SVM (which is the best in [20]) by around 25%. It can be also observed that HCRF-Boost is comparable with the Cardinality Kernel method. Note that the Cardinality Kernel is specialized for MIL. It only induces nonlinearlity to bag classification and still has log-linear models for instance classification. Further, its computational complexity grows quadratically with the number of instances, and needs quadratic space w.r.t. the number of bags. However, HCRF-Boost is a general and flexible method, learns nonlinear potential functions, and provides scalability and efficiency.

[6]We use VLFeat, as in [20, 15], with the same number of codewords as [15] but with fewer codewords than [20] – 1500 for ours but 5000 in [20]). Note that this is not the best setting for the SIFT features. For example, if the codewords are increased to 20,000, the mean average precision increases up to 13.4% by using only the global model. Also by combining or fusing other sets of features, better results can be achieved (e.g. [30, 40]).

Table 4. Comparing our proposed HCRF-Boost with $\propto$SVM algorithms in [20] and the Cardinality Kernel in on TRECVID MED11. The best AP for each event is highlighted in bold.

Event	single–g $\propto$SVM [20]	multi–g $\propto$SVM [20]	Cardinality Kernel [15]	HCRF	HCRF-Boost
6	1.9 %	**3.8** %	2.8 %	1.2 %	2.6 %
7	2.6 %	**5.8** %	**5.8** %	1.8 %	5.3 %
8	11.5 %	11.7 %	17.0 %	9.7 %	**22.4** %
9	4.9 %	5.0 %	**8.8** %	3.0 %	6.3 %
10	0.8 %	0.9 %	**1.3** %	0.8 %	1.1 %
11	1.8 %	2.4 %	3.4 %	1.3 %	**3.7** %
12	4.8 %	5.0 %	10.7 %	4.0 %	**11.3** %
13	1.7 %	2.0 %	**4.7** %	0.8 %	**4.7** %
14	10.5 %	**11.0** %	4.9 %	1.4 %	3.7 %
15	2.5 %	**2.5** %	1.4 %	1.3 %	1.6 %
mAP	4.3 %	5.0 %	6.1 %	2.5 %	**6.3** %

5. Conclusion

We presented a novel and general framework for learning latent structured models. This algorithm uses gradient boosting to train a CRF with hidden variables in functional space. The functional approach helps to learn the structured model directly with respect to the potential functions without direct interaction with the potentially high-dimensional parameter space. By using this method, the potential functions are learned as an ensemble of nonlinear feature functions represented by regression models. This introduces nonlinearity into the model, enhances its feature abstraction and representational power, and finally reduces the chance of overfitting (due to the ensemble effect). We evaluated the performance of the proposed method on three challenging tasks: group activity recognition, human action recognition, and multimedia video event detection. The results showed that our nonlinear ensemble model leads to significant improvement of classification performance compared to the log-linear structured models. Further, the proposed method is very flexible and can be simply integrated with a variety of off-the-shelf nonlinear fitting functions.

References

[1] M. R. Amer, P. Lei, and S. Todorovic. Hirf: Hierarchical random field for collective activity recognition in videos. In *ECCV*, 2014. 2, 6

[2] P. Bühlmann and T. Hothorn. Boosting algorithms: Regularization, prediction and model fitting. *Statistical Science*, pages 477–505, 2007. 4

[3] L.-C. Chen, A. G. Schwing, and R. Urtasun. Learning deep structured models. *ICML*, 2015. 1, 2, 5

[4] T. Chen, S. Singh, B. Taskar, and C. Guestrin. Efficient second-order gradient boosting for conditional random fields. In *AISTATS*, 2015. 2

[5] W. Choi, K. Shahid, and S. Savarese. What are they doing?: Collective activity classification using spatio-temporal relationship among people. In *VS*, 2009. 2, 6

[6] Z. Deng, M. Zhai, L. Chen, Y. Liu, S. Muralidharan, M. J. Roshtkhari, and G. Mori. Deep structured models for group activity recognition. *BMVC*, 2015. 2

[7] T. G. Dietterich, G. Hao, and A. Ashenfelter. Gradient tree boosting for training conditional random fields. *Journal of Machine Learning Research*, 9(10), 2008. 2, 3, 4

[8] T. Do and T. Artieres. Neural conditional random fields. In *AISTATS*, 2010. 2

[9] P. Felzenszwalb, D. McAllester, and D. Ramanan. A discriminatively trained, multiscale, deformable part model. In *CVPR*, 2008. 1, 2

[10] J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001. 3, 4

[11] J. H. Friedman. Stochastic gradient boosting. *Computational Statistics & Data Analysis*, 38(4):367–378, 2002. 5

[12] R. Girshick, J. Donahue, T. Darrell, and J. Malik. Rich feature hierarchies for accurate object detection and semantic segmentation. In *CVPR*, 2014. 1

[13] H. Hajimirsadeghi, M. N. Ahmadabadi, and B. N. Araabi. Conceptual imitation learning based on perceptual and functional characteristics of action. *IEEE Transactions on Autonomous Mental Development*, 5(4):311–325, 2013. 2

[14] H. Hajimirsadeghi, J. Li, G. Mori, M. Zaki, and T. Sayed. Multiple instance learning by discriminative training of markov networks. In *UAI*, 2013. 7

[15] H. Hajimirsadeghi, W. Yan, A. Vahdat, and G. Mori. Visual recognition by counting instances: A multi-instance cardinality potential kernel. *CVPR*, 2015. 2, 7, 8

[16] J. Jancsary, S. Nowozin, T. Sharp, and C. Rother. Regression tree fields – An efficient, non-parametric approach to image labeling problems. In *CVPR*, pages 2376–2383, 2012. 2

[17] S. Khamis, V. I. Morariu, and L. S. Davis. Combining per-frame and per-track cues for multi-person action recognition. In *ECCV*, 2012. 2

[18] A. Krizhevsky, I. Sutskever, and G. E. Hinton. Imagenet classification with deep convolutional neural networks. In *NIPS*, 2012. 1

[19] J. Lafferty, X. Zhu, and Y. Liu. Kernel conditional random fields: representation and clique selection. In *ICML*, 2004. 2

[20] K.-T. Lai, F. X. Yu, M.-S. Chen, and S.-F. Chang. Video event detection by inferring temporal instance labels. In *CVPR*, 2014. 2, 8

[21] T. Lan, Y. Wang, W. Yang, S. N. Robinovitch, and G. Mori. Discriminative latent models for recognizing contextual group activities. *T-PAMI*, 34(8):1549–1562, 2012. 2, 6

[22] W. Li, Z. Zhang, and Z. Liu. Action recognition based on a bag of 3d points. In *CVPRW*, pages 9–14. IEEE, 2010. 2, 6, 7

[23] J. C. Niebles, C.-W. Chen, and L. Fei-Fei. Modeling temporal structure of decomposable motion segments for activity classification. In *ECCV*, 2010. 2

[24] P. Over, G. Awad, J. Fiscus, B. Antonishek, M. Michel, A. F. Smeaton, W. Kraaij, G. Quénot, et al. Trecvid 2011-an overview of the goals, tasks, data, evaluation mechanisms and metrics. In *TRECVID*, 2011. 8

[25] H. Pirsiavash and D. Ramanan. Parsing videos of actions with segmental grammars. In *CVPR*, 2014. 2

[26] A. Quattoni, S. Wang, L.-P. Morency, M. Collins, and T. Darrell. Hidden conditional random fields. *IEEE T-PAMI*, 29(10):1848–1852, 2007. 1, 2, 3, 4

[27] A. Schwing, T. Hazan, M. Pollefeys, and R. Urtasun. Efficient structured prediction with latent variables for general graphical models. In *ICML*, 2012. 2

[28] A. G. Schwing, A. L. Yuille, and R. Urtasun. Fully connected deep structured networks. *arXiv:1503.02351*, 2015. 1, 2, 5

[29] K. Simonyan and A. Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv:1409.1556*, 2014. 1

[30] K. Tang, B. Yao, L. Fei-Fei, and D. Koller. Combining the right features for complex event recognition. In *ICCV*, 2013. 2, 8

[31] Y. Tian, R. Sukthankar, and M. Shah. Spatiotemporal deformable part models for action detection. In *CVPR*, 2013. 2

[32] J. J. Tompson, A. Jain, Y. LeCun, and C. Bregler. Joint training of a convolutional network and a graphical model for human pose estimation. In *NIPS*, 2014. 1, 2

[33] A. Vahdat, K. Cannons, G. Mori, S. Oh, and I. Kim. Compositional models for video event detection: A multiple kernel learning latent variable approach. In *ICCV*, 2013. 2

[34] A. Vedaldi and A. Zisserman. Efficient additive kernels via explicit feature maps. *TPAMI*, 34(3):480–492, 2012. 2

[35] R. Vemulapalli, F. Arrate, and R. Chellappa. Human action recognition by representing 3d skeletons as points in a lie group. In *CVPR*, pages 588–595, 2014. 7

[36] C. Wang, Y. Wang, and A. L. Yuille. An approach to pose-based action recognition. In *CVPR*, 2013. 7

[37] J. Wang, Z. Liu, Y. Wu, and J. Yuan. Mining actionlet ensemble for action recognition with depth cameras. In *CVPR*, 2012. 2, 7

[38] Y. Wang and G. Mori. Hidden part models for human action recognition: Probabilistic versus max margin. *TPAMI*, 33(7):1310–1323, 2011. 2

[39] L. Xia and J. Aggarwal. Spatio-temporal depth cuboid similarity feature for activity recognition using depth camera. In *CVPR*, pages 2834–2841. IEEE, 2013. 2, 7

[40] Z. Xu, I. W. Tsang, Y. Yang, Z. Ma, and A. G. Hauptmann. Event detection using multi-level relevance labels and multiple features. In *CVPR*, 2014. 2, 8

[41] W. Yang, Y. Wang, A. Vahdat, and G. Mori. Kernel latent svm for visual recognition. In *NIPS*, 2012. 2

[42] X. Yang and Y. Tian. Super normal vector for activity recognition using depth sequences. In *CVPR*, 2014. 2, 6, 7

[43] X. Yang, C. Zhang, and Y. Tian. Recognizing actions using depth motion maps-based histograms of oriented gradients. In *ACM MM*, 2012. 2, 7

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

[45] M. Zanfir, M. Leordeanu, and C. Sminchisescu. The moving pose: An efficient 3d kinematics descriptor for low-latency action recognition and detection. In *ICCV*, 2013. 7