Optimizing Non-Decomposable Loss Functions in Structured Prediction

Mani Ranjbar, Tian Lan, Yang Wang, Steven N. Robinovitch, Ze-Nian Li, Greg Mori

Abstract—We develop an algorithm for structured prediction with *non-decomposable* performance measures. The algorithm learns parameters of Markov random fields and can be applied to multivariate performance measures. Examples include performance measures such as F_β score (natural language processing), intersection over union (object category segmentation), Precision/Recall at k (search engines) and ROC area (binary classifiers). We attack this optimization problem by approximating the loss function with a piecewise linear function. The loss augmented inference forms a quadratic program (QP), which we solve using LP relaxation. We apply this approach to two tasks: object class-specific segmentation and human action retrieval from videos. We show significant improvement over baseline approaches that either use simple loss functions or simple scoring functions on the PASCAL VOC and H3D Segmentation datasets, and a nursing home action recognition dataset.

Index Terms—Optimization, Large-Margin, Structural SVM

1 Introduction

We develop an algorithm for learning the parameters of structured Markov Random Field models against non-decomposable performance measures. Solving challenging vision problems such as image understanding, object category segmentation, and video retrieval arguably requires the use of structured models – those incorporating relationships between multiple input and output entities. Evidence for this comes from state-of-the-art approaches to the aforementioned problems. For example, Hoiem et al. [1] formulate image understanding models that tie together object locations, camera parameters, and surfaces. Blaschko and Lampert [2] localize objects using an efficient solution to a structured output regression model. Desai et al. [3] learn models for simultaneously detecting all objects in an image. Non-max suppression and contextual object co-occurrence statistics are learned in a discriminative fashion. Object category segmentation is a canonical example of structured labeling problem – individual pixel labels are not obtained independently, but by considering structured relationships over groups of pixels (e.g. [4], [5], [6]).

For many of these problems the natural performance measures are also "non-decomposable" – ones that do not decompose into a simple sum of individual terms measured over each output entity. Examples of such measures are object detection scores that penalize for multiple detections on a single true positive (e.g PASCAL VOC [7]) and region labeling or object segmentation scores that penalize for over and under labeling or segmentation (e.g. intersection / union score). Typical methods for solving these problems learn parameters against other performance measures, e.g. Hamming loss for segmentation, and then apply post-processing techniques (e.g. non-maximum suppression in object detection) to address the structure in the performance measure. However, these methods fail to take into account structural properties such as connectivity of the variables or counts of the variables with a certain value (e.g. they do not correctly handle multiple detections or tradeoffs in over/under segmentation). We argue that directly optimizing against the non-decomposable loss is superior to these post-processing approaches. Hence, in this paper we develop an algorithm for linking these two together and formulate learning as jointly considering the complex, structured relationships between output variables in the model and in the learning objective.

The main contribution of this paper is developing a general algorithm for addressing this type of learning problem with non-decomposable models and those non-decomposable loss functions which are a function of false positive and false negative counts. We specifically apply it to two problems, object category segmentation and human action retrieval, but note that the algorithm can be applied more broadly. We experiment with Markov Random Field (MRF) models. For segmentation, this is a standard model that contains both unary terms for labeling pixels and pairwise terms on the labels of adjacent pixels. For action retrieval, we formulate a novel MRF that can capture contextual relationships between the actions of the people in a scene. In both cases, we show that learning the parameters to the model under an objective directly tied to the performance measure

- *M. Ranjbar, T. Lan, Z. N. Li and G. Mori are with the School of Computing Science, Simon Fraser University, Burnaby, BC, V5A1S6. E-mail: mra33@sfu.ca, tla58@sfu.ca, li@cs.sfu.ca, mori@cs.sfu.ca*
- *Y. Wang is with the Computer Science Department, Univ. of Illinois at Urbana-Champaign. E-mail: kywang@gmail.com*
- *S. N. Robinovitch is with the School of Kinesiology, Simon Fraser University, Burnaby, BC, V5A1S6. E-mail: stever@sfu.ca*

significantly improves performance relative to baseline algorithms.

This paper builds on our preliminary work [8]. In this paper we formulate a multi-label version of the method, with different inference scheme, and new experiments on object category segmentation and action retrieval.

2 Previous Work

A wide range of learning algorithms exist. Despite technical differences, all of these approaches rely on a performance measure to define what is a "good" result. Based on the complexity of the performance measure, two general approaches to optimize it are imaginable, formulate the learning problem to directly optimize this measure, or approximate this measure with a simpler one and try to optimize it aiming to indirectly optimize the original non-decomposable performance measure. We will call the former "direct optimization" and the latter "indirect optimization".

Due to the complexity of some performance measures, e.g., average precision and intersection over union, many state-of-the-art approaches in different challenges exploit an indirect optimization. Looking at the PASCAL VOC challenge 2010 [7], for example, average precision and intersection over union are defined as performance measures for detection and segmentation tasks respectively, but methods for both tasks use indirect optimizations for solving these problems.

Structured models are arguably a requirement for robust solutions to learning problems in a variety of application domains. Tasks such as machine translation, object category segmentation, and scene understanding involve reasoning about relationships between words in a document, pixels in an image, and objects in a scene respectively. In addition, the performance measures for these applications often are non-decomposable and are not a simple sum of terms measured over individual output entities. Instead, they measure performance as a function of the entire, structured output. The focus of this paper is developing a learning approach that can handle these together, handling structured prediction while optimizing against certain non-decomposable performance measures.

Modeling dependencies between outputs while optimizing against a loss function has been a research topic for many years. Optimizing the expected loss in this scenario is a non-convex problem. However, Taskar et al. [9] and Tsochantaridis et al. [10] have proposed rather to optimize a convex relaxation of the expected loss. The cutting-plane algorithm has been shown to be efficient for solving this optimization [10]. Teo et al. [11] presented a bundle method, which is basically the cutting-plane method stabilized with Moreau-Yosida regularizer and prove a tighter bound on the duality gap. Taskar et al. [12] solves the same problem using the extragradient method. Extragradient consists of a gradient descent followed by a projection to the feasible set. Shalev-Shwartz et al. [13] proposed Pegasos, which works solely in the primal space. Similar to [12], Pegasos consists of a gradient descent step followed by a projection step. The computational difficulty in all aforementioned structured prediction approaches is finding the subgradient, which requires solving the "most violated constraint" [10] or "loss augmented inference" [14]. It is shown that for decomposable performance measures learning is tractable when the model is a submodular MRF or a matching [9], [10], [12]. In contrast, in this paper we focus on non-decomposable performance measures.

Joachims [15] proposed an approach to efficiently compute the most violated constraint for a large class of non-decomposable loss functions, a subset of those we consider in this paper. However, the underlying models were limited, and do not permit pairwise interactions between output labels. The method of Yue et al. [16] takes a similar approach to optimize against Mean Average Precision. Khanna et al. [17] present an algorithm in the same framework to optimize against normalized discounted cumulative gain (NDCG). Rather than solving a convex relaxation of the expected loss, McAllester et al. [18] proposed a perceptron-like training approach to directly optimize the original loss function, but still need to solve the loss augmented inference. For the problems in which the inference procedure is not tractable, Finley et al. [19] compare under-generating and over-generating algorithms in structured prediction and conclude that "overgenerating methods [LP and graph cut] have theoretic advantages over undergenerating [LBP, greedy] methods".

In this paper we provide an algorithm for structured prediction with a non-decomposable scoring function that optimizes against non-decomposable performance measures, those which are a function of false positive and false negative counts.

3 Background

To create a foundation for the proposed approach, we start with an overview of our learning formulation. Next, we discuss the two common approaches, one based on decomposable loss functions with non-decomposable scoring functions and the others with non-decomposable loss functions and decomposable scoring functions. We call a loss function simple if it can be decomposed into loss on individual training samples. Likewise, a scoring function is called simple if it only depends on a single sample point and its ground-truth label. Finally, we propose a framework to incorporate certain non-decomposable loss functions and non-decomposable scoring functions in structured prediction.

For notational convenience, we write matrices with bold upper case letters (e.g. $\boldsymbol{X}$), vectors with bold lower case letters (e.g. $\boldsymbol{x}$) and scalars with normal lower case letters (e.g. x). In our notation, $\boldsymbol{x}_i$ represents the i^{th} column of matrix $\boldsymbol{X}$ and x_j represents the j^{th} element of vector $\boldsymbol{x}$. We use superscripts to denote variables or vectors that do not belong to a vector or a matrix (x^i, $\boldsymbol{x}^i$).

3.1 Problem Formulation

The goal of our learning problem is defined as finding a function $h \in \mathcal{H}$ from the hypothesis space $\mathcal{H}$ given training samples S = (($\boldsymbol{x}^1$, y^1), ...,($\boldsymbol{x}^N$, y^N)) that optimizes the expected prediction performance on the new samples S' of size N'.

$$R^{\Delta}(h) = \int \Delta\Big(\big[h(\boldsymbol{x}'^1), h(\boldsymbol{x}'^2), \ldots, h(\boldsymbol{x}'^{N'})\big], \big[y'^1, y'^2, \ldots, y'^{N'}\big] \Big) dPr(S'). \quad (1)$$

In general, the loss function Δ cannot be decomposed into a linear combination of a loss function δ over individual samples. But, for simplicity, most discriminative learning algorithms (e.g. SVM) assume decomposibility and i.i.d. samples, which allows for rewriting Eq. 1 as

$$R^{\Delta}(h) = R^{\delta}(h) = \int \delta(h(\boldsymbol{x}'), y') dPr(\boldsymbol{x}', y'). \quad (2)$$

Instead of solving the estimated risk in Eq. 2, learning algorithms approximate that with empirical risk $\hat{R}^{\delta}$ defined as

$$\hat{R}^{\delta}(h) = \frac{1}{N} \sum_{i=1}^{N} \delta(h(\boldsymbol{x}^i), y^i). \quad (3)$$

For non-decomposable loss functions, such as F_1 score or intersection over union, optimizing Eq. 2 does not provide the desired answer. Rather, we are interested in finding an algorithm that can directly optimize the empirical risk based on the sample loss,

$$\hat{R}^{\Delta}_{S}(h) = \Delta\left((h(\boldsymbol{x}^1), \ldots, h(\boldsymbol{x}^N)), (y^1, \ldots, y^N)\right). \quad (4)$$

Note that finding an $h \in \mathcal{H}$ that optimizes Eq. 4 for an arbitrary loss function Δ can be computationally challenging.

3.2 Structured Prediction Learning

For non-decomposable loss functions, one can reformulate the SVM based on the idea of multivariate prediction [15]. Instead of having a mapping function $h : \mathcal{X} \rightarrow \mathcal{Y}$ from a single example $\boldsymbol{x}$ to its label y, where $\boldsymbol{x} \in \mathcal{X}$ and $y \in \{-1, +1\}$, we look at all examples at once and try to learn a mapping function $\bar{h} : \mathcal{X} \times \cdots \times \mathcal{X} \rightarrow \bar{\mathcal{Y}}$, where $\bar{\mathcal{Y}} = \{-1, +1\}^N$. We define $\boldsymbol{X} = [\boldsymbol{x}^1, \ldots, \boldsymbol{x}^N]$, and $\boldsymbol{y} = [y^1, \ldots, y^N]$.

We can define the best labeling using a linear discriminant function (scoring function)

$$\bar{h}(\boldsymbol{X}) = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \boldsymbol{w}^T \Psi(\boldsymbol{X}, \boldsymbol{y}'). \quad (5)$$

Here, function Ψ measures the compatibility of the data points and their assigned labels. If we define the Ψ function as a simple form

$$\Psi(\boldsymbol{X}, \boldsymbol{y}') = \sum_{i=1}^{N} y'_i \boldsymbol{x}_i, \quad (6)$$

that only depends on individual training points and their labels, the optimal labeling sequence would be

$$\arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \boldsymbol{w}^T \Psi(\boldsymbol{X}, \boldsymbol{y}') = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \sum_{i=1}^{N} y'_i \boldsymbol{w}^T \boldsymbol{x}_i \quad (7)$$

$$= \left(h(\boldsymbol{x}^1), \ldots, h(\boldsymbol{x}^N)\right), \quad (8)$$

which is exactly the same as the optimal labeling in SVM.

One way of incorporating a loss function Δ in SVM formulation is *Margin Rescaling*[9],

$$\min_{\boldsymbol{w}, \xi \geq 0} \|\boldsymbol{w}\|^2 + C\xi \quad (9)$$

$$s.t.\ \forall \boldsymbol{y}' \in \bar{\mathcal{Y}} \backslash \boldsymbol{y},\ \boldsymbol{w}^T \left[\Psi(\boldsymbol{X}, \boldsymbol{y}) - \Psi(\boldsymbol{X}, \boldsymbol{y}')\right] \geq \Delta(\boldsymbol{y}, \boldsymbol{y}') - \xi$$

Similar to the original SVM formulation, ξ in Eq. 9 is an upper bound on $\Delta(\bar{h}(\boldsymbol{X}), \boldsymbol{y})$[15].

The guarantee for convergence in polynomial time, the potential for incorporating complex loss functions in the objective and good performance in practice are the most important reasons why structured prediction has garnered much attention in computer vision recently.

In the standard approaches for solving Eq. 9, the output vector, $\tilde{\boldsymbol{y}}$, corresponding to the most violated constraint should be found repeatedly [20],

$$\tilde{\boldsymbol{y}} = \arg\max_{\boldsymbol{y}' \in \bar{\mathcal{Y}}} \Delta(\boldsymbol{y}, \boldsymbol{y}') + \boldsymbol{w}^T \Psi(\boldsymbol{X}, \boldsymbol{y}'). \quad (10)$$

Finding $\tilde{\boldsymbol{y}}$ is computationally challenging given an arbitrary loss function, $\Delta(\boldsymbol{y}, \boldsymbol{y}')$, and compatibility function, $\Psi(\boldsymbol{X}, \boldsymbol{y}')$. However, solving Eq. 10 in two special cases has been shown to be efficient. We categorize these approaches based on the simplicity of their Δ and Ψ functions. We call a loss function simple if it can be decomposed into individual training samples. Likewise, a compatibility function is called simple if it decomposes over single sample points and their ground-truth labels.

3.3 Decomposable Δ, Complex Ψ

Optimizing the parameters of a MRF structure when the loss function can be decomposed into the loss of individual samples falls into this category [9]. One popular application in this category is foreground-background segmentation with Hamming loss, which is defined as

$$\Delta_H = \sum_i \mathbb{1}_{[y_i \neq y'_i]}. \quad (11)$$

where, $\mathbb{1}_{[]}$ is the indicator function. Szummer et al. [6] have employed this formulation and reported promising results for interactive segmentation.

Decomposibility of the loss function results in a MRF form for Eq. 10, because the loss function can be treated as another unary term that adds up to the unary terms of the compatibility function. Assuming binary labels, this MRF can be solved efficiently using graphcut.

The advantage of this approach is to exploit pairwise connections, but it is only tractable for decomposable loss functions.

3.4 Non-decomposable Δ, Simple Ψ

The other special case presented by Joachims [15], is when the Ψ function has a simple form of

$$\Psi(\boldsymbol{X}, \boldsymbol{y}') = \sum_{i=1}^{N} y_i' \boldsymbol{x}_i. \tag{12}$$

If the loss function, Δ, is just a function of true positive (TP), false positive (FP) and false negative (FN), then there are at most $N_p \times N_n$ distinct loss values, where N_p and N_n represent the number of positive and negative training examples, respectively. Hence, Eq. 10 can be solved by iterating over all loss values and maximizing $\boldsymbol{w}^T \Psi(\boldsymbol{X}, \boldsymbol{y}')$ subject to the value of TP, FP and FN [15].

Unlike the approach of Taskar et al. [9], many standard accuracy measures that lead to non-decomposable loss functions, such as F_β score (natural language processing), intersection over union (object category segmentation), Precision/Recall at k (web search engines) and ROC area (binary classifiers) can be directly optimized by this approach. However, this method cannot benefit from the pairwise interactions of training samples, which are shown to be advantageous in many applications, such as object detection [3] and scene interpretation [1].

4 Proposed Approach: Solving Non-decomposable Δ, Complex Ψ

Discussing the advantages and shortcomings of the previous methods, we now propose an approach to directly optimize certain complex loss functions in a MRF. Here, we can optimize non-decomposable accuracy measures, such as F_β and intersection over union and still be able to benefit from pairwise interactions between training points. So far, we have considered only binary output problems for simplicity, but for the rest of the paper the output is assumed to be multilabel. For notational convenience, we encode the output label, $\boldsymbol{y}$ in one-of-M format, where $M = |\mathcal{L}|$ and $\mathcal{L}$ is the set of all possible labels. In this encoding, the assigned label of y_i is represented using a binary vector of size M such that its j^{th} element is 1, when the j^{th} label is assigned to this output, and the rest of its elements are 0.

We choose to follow the general framework of Structural$_{\text{SVM}}$ [20], shown in Eq. 9. Solving Eq. 9 requires finding the most violated constraint (Eq. 10) at each iteration and modifying the parameter vector $\boldsymbol{w}$ accordingly. We propose a novel method to efficiently solve for an approximate most violated constraint for certain non-decomposable loss functions in presence of pairwise terms in the compatibility function, Ψ.

We can summarize the proposed approach as

1) Replacing the original non-decomposable loss function with a piecewise linear approximation,
2) Writing the problem of finding the most violated constraint as a quadratic program,
3) Converting the quadratic program to a linear program and solve the relaxed problem.

4.1 Piecewise Linear Approximation

Many standard accuracy measures, including the one presented in the previous section, share the property that they can be computed from the contingency table[1]. Given the number of positive and negative examples, N_p and N_n, the loss function corresponding to these accuracy measures is just a function of FP and FN. Using piecewise linear approximation, we can write

$$\begin{aligned}\Delta(FP, FN) &\simeq \tilde{\Delta}(FP, FN) \\ &= \sum_{r=1}^{Q} \mathbb{1}_{[(FP,FN)\in\mathfrak{R}_r]} \{\alpha_r FP + \beta_r FN + \gamma_r\}\end{aligned} \tag{13}$$

where, Q is the number of subregions (pieces), α_r, β_r and γ_r represent the r^{th} plane coefficients and $\mathfrak{R}_r$s are the subregions that partition the space spanned by FP and FN.

As an example, Figure 1 illustrates the intersection over union loss function,

$$\Delta_{\frac{\cap}{\cup}}(FP, FN) = \frac{FN + FP}{N_p + FP}, \tag{14}$$

along with its piecewise linear approximations using 15 and 40 pieces.

Given the subregion $\mathfrak{R}_r$, the original non-linear loss function is a linear function of FP and FN. The next step is to substitute the approximated loss function, $\tilde{\Delta}$ into Eq. 10 and solve for the most violated constraint.

4.2 Forming the Quadratic Program

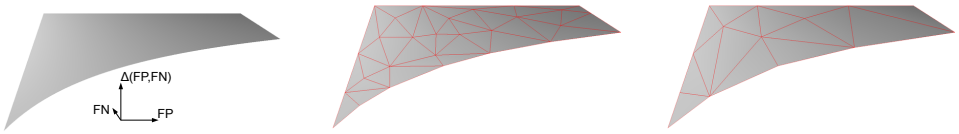
Capturing the structure of the output requires a model that is rich enough to absorb the dependencies between the outputs. At the same time, a preferred model candidate offers tractable inference procedure. A choice that satisfies both requirements are MRFs, which are commonly used for modeling interdependent inputs and outputs in many applications.

We assume that we are given a MRF represented by a graph $G = (V, E)$ where V is the set of nodes with $N = |V|$, and E is the set of edges. The output label takes value from the set $\mathcal{L}$, which has M members. We define our Ψ with unary and pairwise terms as

$$\begin{aligned}\Psi(\boldsymbol{X}, \boldsymbol{Y}) = &\sum_{i=1}^{N}\sum_{k=1}^{M} y_{ik}\phi_u(\boldsymbol{x}_i) + \\ &\sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\sum_{k=1}^{M}\sum_{l=1}^{M} y_{ik}y_{jl}\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j).\end{aligned} \tag{15}$$

Here $\mathcal{N}_i$ is the set of neighbors of node i. We later explain how we define the unary and pairwise features (ϕ_u and ϕ_p) in our experiments. We rewrite Eq. 10 with

1. Are just a function of TP, FP, TN and FN.



a b c

Fig. 1. Intersection over union loss surface in FP and FN space. a) Exact surface, b) a piecewise linear approximation with 40 subregions, c) a piecewise linear approximation with 15 subregions.

approximated loss function, $\tilde{\Delta}$ as

$$\tilde{\boldsymbol{Y}}^* = \arg\max_{\boldsymbol{Y}'} \tilde{\Delta}(\boldsymbol{Y}, \boldsymbol{Y}') + \boldsymbol{w}^T \Psi(\boldsymbol{X}, \boldsymbol{Y}') \tag{16}$$

$$= \arg\max_{\boldsymbol{Y}'} \tilde{\Delta}(\boldsymbol{Y}, \boldsymbol{Y}') + \overset{u}{\boldsymbol{w}}^T \sum_{i=1}^{N}\sum_{k=1}^{M} y'_{ik}\phi_u(\boldsymbol{x}_i) + \overset{p}{\boldsymbol{w}}^T \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\sum_{k=1}^{M}\sum_{l=1}^{M} y'_{ik}y'_{jl}\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j) \tag{17}$$

where $\boldsymbol{w} = [\overset{u}{\boldsymbol{w}}; \overset{p}{\boldsymbol{w}}]$ (weights of the unary features $\overset{u}{\boldsymbol{w}}$ concatenated with the weights of the pairwise features $\overset{p}{\boldsymbol{w}}$).

The group of non-decomposable loss functions that are considered in the proposed approach are a function of false positive and false negative counts. Although the definition of false positive and false negative counts are straight forward in binary output problems, for multi-label problems such definitions are task-dependent. In this paper, we assume that the loss is defined for one label (label p) versus the rest and therefore, define the false positive and false negative counts as

$$FP_{\boldsymbol{Y},\boldsymbol{Y}'} = \sum_{i=1}^{N} y'_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y_{ik}, \tag{18}$$

$$FN_{\boldsymbol{Y},\boldsymbol{Y}'} = \sum_{i=1}^{N} y_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y'_{ik}. \tag{19}$$

Assuming that the loss values fall in subregion $\mathfrak{R}_r$, we can write Eq. 17 as

$$\tilde{\boldsymbol{Y}}^* = \arg\max_{\boldsymbol{Y}'} \Bigg(\alpha_r \sum_{i=1}^{N} y'_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y_{ik} + \beta_r \sum_{i=1}^{N} y_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y'_{ik} + \gamma_r + \overset{u}{\boldsymbol{w}}^T \sum_{i=1}^{N}\sum_{k=1}^{M} y'_{ik}\phi_u(\boldsymbol{x}_i) + \overset{p}{\boldsymbol{w}}^T \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\sum_{k=1}^{M}\sum_{l=1}^{M} y'_{ik}y'_{jl}\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j) \Bigg). \tag{20}$$

Note that Eq. 20 only includes the predicted label y' in linear and quadratic forms. Hence, we can write a quadratic program based on Eq. 20 subject to the loss values being in subregion $\mathfrak{R}_r$.

$$\Bigg(\overbrace{\sum_{i=1}^{N} y'_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y_{ik}}^{FP}, \overbrace{\sum_{i=1}^{N} y_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y'_{ik}}^{FN} \Bigg) \in \mathfrak{R}_r \tag{21}$$

In order to have linear constraints in Eq. 21, the boundary of all subregions should be definable as a linear function of $\boldsymbol{y}'$. One way is to separate the subregions by straight lines. If for example, we partition the space spanned by FP and FN into triangles (Fig. 1-b,c) then Eq. 21 will be substituted by three linear constraints corresponding to the three sides of the triangle.

4.3 Converting Quadratic Program to Linear Program

The quadratic function in Eq. 20 is potentially non-convex, since there is no constraint on the coefficients of this function. So, instead of looking for a local optima of this non-convex function, we relax the problem (MAP-MRF LP relaxation [21]) by introducing some variables that substitute the quadratic terms in the this function and form a linear program, which is convex. In detail, we introduce $\mathfrak{y}'^{ij}_{kl} = y'_{ik}y'_{jl}$. To relate these new variables to the output variables $\boldsymbol{y}'$, we augment some linear inequality constraints in the form, $\mathfrak{y}'^{ij}_{kl} \leq y'_{ik}$, $\mathfrak{y}'^{ij}_{kl} \leq y'_{jl}$ and $\sum_{k,l} \mathfrak{y}'^{ij}_{kl} = 1$. The final linear program that needs to be solved for subregion $\mathcal{R}_r$ is

Maximize:

$$\alpha_r \sum_{i=1}^{N} y'_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y_{ik} + \beta_r \sum_{i=1}^{N} y_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y'_{ik} + \gamma_r + \overset{u}{\boldsymbol{w}}^T \sum_{i=1}^{N}\sum_{k=1}^{M} y'_{ik}\phi_u(\boldsymbol{x}_i) + \overset{p}{\boldsymbol{w}}^T \sum_{i=1}^{N}\sum_{j\in\mathcal{N}_i}\sum_{k=1}^{M}\sum_{l=1}^{M} \mathfrak{y}'^{ij}_{kl}\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$$

Subject to: (22)

$$\Bigg(\sum_{i=1}^{N} y'_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y_{ik}, \sum_{i=1}^{N} y_{ip} \sum_{k\in\{1,\dots,M\}\setminus p} y'_{ik} \Bigg) \in \mathfrak{R}_r$$

$$\mathfrak{y}'^{ij}_{kl} \leq y'_{ik},\ \mathfrak{y}'^{ij}_{kl} \leq y'_{jl}$$

$$\sum_{k,l} \mathfrak{y}'^{ij}_{kl} = 1,\ \sum_{k} y'_{ik} = 1$$

$$y'_{ij}, \mathfrak{y}'^{ij}_{kl} \in \{0,1\}, \quad i \in \{1,\dots,N\},\ j \in \mathcal{N}_i,\ k,l \in \mathcal{L}$$

Solving this LP for thousands of binary variables (labels), is not computationally tractable. So instead we relax the

label values to real numbers between zero and one and solve for optimal labeling. Later, we map the optimal labels to integer values, if necessary, by rounding the results. We solve Eq. 22 for each subregion separately, and return the labeling of the one with the maximum objective value as the most violated constraint.

4.4 Inference

The inference procedure concerns about maximizing the assignment score $\boldsymbol{w}^T\Psi(\boldsymbol{X},\boldsymbol{Y})$ over different assignments and is MAP-MRF problem. For general multilabel problems this task is shown to be NP hard (see [5]), but many approximate inference approaches has been proposed to solve this problem [21], [22]. However, for a supermodular binary problem efficient min-cut/max-flow algorithms exist that can solve the inference exactly [23].

For many applications, not only the maximizing assignment, but also the confidence score associated to each element's assignment is required. Given the confidence scores, compromising FP versus FN or vice versa is possible.

One way of computing a confidence score for the i^{th} element of the output is by looking at the difference in the scores, when the i^{th} output is assigned to the positive label p comparing to when it is assigned to any other label, while the rest of the output elements get their best assignments [3]. Formally,

$$s_i = \max_{\boldsymbol{Y},y_{ip}=1} \boldsymbol{w}^T\Psi(\boldsymbol{X},\boldsymbol{Y}) - \max_{\boldsymbol{Y},y_{ip}\neq 1} \boldsymbol{w}^T\Psi(\boldsymbol{X},\boldsymbol{Y}) \quad (23)$$

Each maximization in Eq. 23 is a MAP estimate in the MRF. For our segmentation experiments, we only require the best labeling, but for our action retrieval experiment the score is needed to compare different approaches.

5 Experiments

To highlight the superiority of the proposed approach over two existing alternatives – keeping the model non-decomposable, but optimizing against a decomposable loss function, or keeping the loss function non-decomposable, but forgetting about the dependency between outputs and employing decomposable models – we design a set of experiments. We compare the methods on two applications – object category segmentation and action retrieval.

5.1 Learning Method

We utilize NRBM [24] – an instance of a bundle method – as the core of our learning and solve the loss augmented inference based on the proposed approach. NRBM solves the unconstrained form of Eq. 9

$$\min_{\boldsymbol{w}} \frac{\lambda}{2}\|\boldsymbol{w}\|^2 + \max_{\boldsymbol{Y}'} \Big(\boldsymbol{w}^T\Psi(\boldsymbol{X},\boldsymbol{Y}') + \Delta(\boldsymbol{Y}',\boldsymbol{Y})\Big) - \boldsymbol{w}^T\Psi(\boldsymbol{X},\boldsymbol{Y}) \quad (24)$$

Note that other structured prediction formulations such as Pegasos [13] or the formulation proposed by Meshi et al. [25] could easily replace the bundle method. We chose NRBM due to implementation simplicity knowing that it has the same bound of $O(1/\epsilon)$ like aforementioned alternatives to obtain a solution of accuracy ϵ. We do cross validation to set the λ parameter in Eq. 24. To be able to solve the inference as well as the loss augmented inference exactly, for the first set of experiments involving binary output labellings, we add $\overset{p}{\boldsymbol{w}} \preccurlyeq 0$ constraints similar to Szummer et al. [6] to make the function supermodular.

5.2 Mesh Creation

The main idea of this paper is to approximate the loss function with a piecewise planar function in false positive and false negative space, in which the loss function is assumed to live. The process of computing the piecewise planar approximation is offline and could be performed using many approaches. We choose to start with a dense mesh in false positive and false negative space and employ mesh simplification methods to reduce the number of pieces to the desired number. To keep the domain of the loss function intact, the boundary of the original mesh and the simplified version should be the same. The minimum number of pieces that a mesh can be simplified to before violating this property is dictated by the simplification method and the primitive shapes that create the mesh.

The other alternative for creating the approximate mesh is to fix the number of vertices and initialize the approximate surface, e.g. on a grid. Then, try to minimize the distance between the original and approximated meshes. The distance function is non-convex in most cases and it is hard to find its global minimum. Instead, general techniques such as gradient descent could be employed to find a local minimum of this function.

In our experiments, we use MeshLab [26] and set the number of pieces to 15, which is the lowest that respects the boundary condition for all of the loss surfaces in the experiments. We tried higher numbers of pieces, but did not notice significant improvement in the overall accuracy. An example of an original densely created loss function along with its approximated versions are shown in Fig 1. We use "quadric edge collapse " technique in MeshLab, which simplifies the mesh based on the method of Gerland et al. [27]. The approximated surface resembles the original loss surface quite accurately. For example, the average absolute distance and the maximum absolute distance between the approximated surface and the original surface are 0.0011 and 0.031, respectively in Pascal VOC 2009 dataset when the loss value varies between zero and one.

5.3 Baseline Methods

We implement two baseline approaches to compare against, each including one aspect of our proposed method. The first baseline, which we name "Hamming", consists of our model, but optimized against Hamming

loss, a decomposable loss function that is used widely for structured prediction [9], [10], [6]. Hamming loss is defined as

$$\Delta_{\text{Hamming}} = \pi_1 FP + \pi_2 FN \tag{25}$$

where, π_1 and π_2 adjust the contribution of FP and FN in the overall loss. We set $\pi_2 = 1/2N_p$ and cross validate the ratio π_2/π_1 on the set {1, 2, 5, 10, 50, 100, N_n/N_p} (which is a time-consuming process). Here, N_p and N_n represent the number of positive and negative examples in the training set. Solving the loss augmented inference given this loss function is as hard as solving the inference problem, because the loss is augmented to each node in the graph as a unary term. Comparison to this baseline reveals the importance of the proposed learning framework, which lets us optimize against non-decomposable loss functions.

To show the importance of the structure in the model (smoothing in segmentation and intra-frame and inter-frame interactions in action retrieval), we implement the approach of Joachims [15]. This approach can exactly optimize against multivariate non-decomposable performance measures, the ones that can be approximately optimized using the proposed approach, but only for decomposable models. We remove the pairwise interactions from the model and train the model parameters using only the unary features. We call this approach "Unary" in the results.

5.4 Object Category Segmentation

We employ object category segmentation as an example of a structured output problem with binary outputs. The task is to label the pixels of an image as being part of a known object (foreground) or not (background). We set the label of foreground to one and the label of background to zero. Intersection over union, measured over the entire dataset of images, is used to compare object category segmentation accuracies on the Pascal VOC challenge [7]. It is defined as

$$Acc_{\frac{\cap}{\cup}}(FP, FN) = \frac{N_p - FN}{N_p + FP} \quad \Longleftrightarrow \quad \Delta_{\frac{\cap}{\cup}}(FP, FN) = \frac{FP + FN}{N_p + FP}, \tag{26}$$

We optimize against this loss function and compare to the baselines on three datasets – Pascal VOC 2009, Pascal VOC 2010 and H3D. Solving the MAP inference and the loss augmented inference exactly requires a supermodular scoring function. So, for this experiment we guarantee supermodularity by forcing the weights corresponding to pairwise features to be negative, knowing that the pairwise features are always positive.

5.5 Pixels vs. Superpixels

If we decide to perform segmentation on the pixel level, meaning that the input be the set of all features extracted from all pixels in the dataset and the output be the binary label of each pixel, then for Pascal VOC 2009 dataset we would have $133,567,772$ pixels and the same number of nodes in our MRF. In average each node in our graph has about 4 neighbors, which would create around 535×10^6 edges in the graph. Learning the parameters on this huge graph is intractable both for the baseline methods and the proposed approach. Moreover, the features extracted from a group of nearby similar pixels are perhaps more robust comparing to the features extracted from single pixels. As an alternative, nearby pixels that share similar appearance features could form a group (superpixel) and share the same label. The down side of moving from pixel to superpixel is the possibility that the pixel of a superpixel come from both foreground and background. In this case, the maximum achievable accuracy drops.

TABLE 1
Maximum achievable accuracy percentage in VOC 2009, VOC 2010 and H3D datasets due to superpixelization.

	Aeroplane	Bus	Car	Horse	Person	TV/Monitor
VOC09	73.17	85.34	77.2	66.48	74.88	86.2
VOC10	72.63	82.24	77.67	68.37	74.06	85.78
H3D	-	-	-	-	79.11	-

In our experiments we employ the superpixel extractor of Felzenszwalb et al. [28] and set its parameters to $MinArea = 2000, k = 200, \sigma = 0.01$. This setting of parameters result in an average of 50 superpixels per image of size 300×500 pixels. Using this parameters the number of nodes in the graph decreases from about 134×10^6 to approximately 27×10^3 nodes in Pascal VOC 2009 dataset and from about 190×10^6 to approximately 24×10^3 nodes. However, as explained before, the maximum possible accuracy drops from 100% to the numbers reported in Table 1.

The second baseline assumes that all positive examples (superpixels of the foreground) contribute equally in the loss function, which is not true if the area of the superpixels are different[2]. On the other hand, this approach has $O(N^2)$ complexity, when N is the number of nodes in the graph and clearly is not tractable if working on pixels. Assuming the same features for each pixel of a superpixel, we have modified Joachims [15] algorithm to work on superpixels as follows. Instead of sorting the superpixels based on their scores, we sort superpixels by their scores divided by their areas. We also adjust the value of the loss function based on the area of the superpixels. This approach is guaranteed to produce correct labellings for all superpixels except possibly one foreground and one background superpixels[3].

2. The other alternative is to force the superpixels to have the same size, but then large flat regions such as sky would be broken into many small superpixels and regions of small objects could be grouped with background regions.

3. Algorithm 2 of [15] has been proven to find the optimal assignment if all positive examples contribute equally in the loss function and also do all negative examples. Based on Algorithm 2 of [15], the first a positive examples get value 1 and the rest get value 0 at the optimal v. Knowing that the pixels of a superpixel are sorted sequentially, the only superpixel that may have inconsistent labels is the one that its pixel is located at position a. The same argument holds for background superpixels. So, all pixels of other superpixels get the same labels as they would get if we could afford to run the algorithm on pixels.

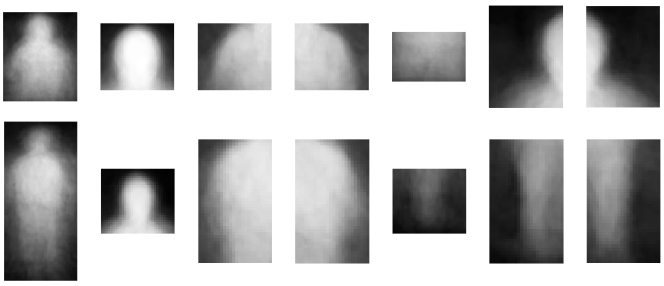


Fig. 2. Visualization of the average root and part shapes in person category. Each row corresponds to shape models obtained from root and part appearance models of one object pose.

5.5.1 Features

We define an MRF segmentation model with unary and pairwise features, for which the exact inference is performed using min-cut/max-flow algorithm [23].

In the MRF, there is an edge between each pair of adjacent superpixels i and j. This is a standard method to model label smoothness in each image. We define a set of pairwise features that represent $\phi_p(\boldsymbol{x}_i, \boldsymbol{x}_j)$ in Eq. 22. We first convert the image from RGB to La^*b^* color space. We define L_i, a_i and b_i to be the average L, a and b values inside superpixel i, respectively and assign the length of the common boundary between superpixel i and j to $\mathcal{P}_{ij}$. We then compute the pairwise features as

$$\boldsymbol{\phi}_p(\boldsymbol{x}_i, \boldsymbol{x}_j) = \mathcal{P}_{ij}. \exp\left[-\tau_1(L_i - L_j)^2, -\tau_2(a_i - a_j)^2, -\tau_2(b_i - b_j)^2\right] \quad (27)$$

In our experiments the values of τ_1 and τ_2 are set to 2×10^{-2} and 5×10^{-3}, respectively.

To represent each superpixel, we use a set of bottom-up and top-down features, which form the unary features $\phi_u(\boldsymbol{x}_i)$ for superpixel i in Eq. 22. To create the bottom-up features, we compute Color SIFT features [29] on a dense grid with 6 pixel spacing in horizontal and vertical directions. We then turn this into a bag-of-words representation using a codebook of 1000 visual words.

For top-down features, we take a similar approach to the implicit shape model [30]. We first learn two appearance models for each of the 6 object categories using the detector of Felzenszwalb et al. [31]. The result includes two root filters and 6×2 part filters, where each root filter and 6 corresponding part filters model the object appearance in one pose. We run this detector on the training set and collect all bounding boxes that have positive scores. We then crop the ground-truth images on the bounding box locations and compute the average shape for the roots and parts, Fig 2.

We explain the rest of the process for one part, but the same process is applied to all parts and both roots. We find the potential part locations and their confidences by running the detector on the image in different scales. We call the result at each scale a confidence map, Fig. 3-b.

TABLE 2
Background to foreground pixel ratio in Pascal VOC 2009 and 2010

	Aeroplane	Bus	Car	Horse	Person	TV/Monitor
VOC 2009	163	69	86	130	24	96
VOC 2010	168	64	70	119	26	105

Each potential part location casts its vote for the shape of that part proportional to its confidence. We implement this by convolving the confidence maps (different scales) with the average shape for that particular part. We call the convolution result in each scale a potential mask, Fig. 3-c. To merge the potential masks, we rescale them to the original image size and get the maximum of the masks, Fig. 3-d. We accumulate the mask values inside each superpixel to form the top-down feature corresponding to the part. Fig. 3 depicts the entire process for one part.

5.5.2 Pascal VOC 2009 and 2010 Segmentation Datasets

The Pascal VOC 2009 dataset includes 749 pixel-level labeled training images and 750 validation images. The Pascal VOC 2010 dataset includes 964 training and 964 validation images. We decide to train our method on the training set and test on the validation set, because the ground-truth for the test set is not publicly available and our focus is on comparison to baseline methods using a different model or learning criterion. We present the results on 6 object categories, Aeroplane, Bus, Car, Horse, Person, and TV/Monitor. We select these categories because the top-down unary features obtained from the Felzenszwalb et al. object detector [31] provide reasonable detection on them. Without the top-down features, the overall accuracy would be so low as to make the comparison between different learning methods uninformative. Note that we perform the experiments on these objects independently. For example, when we segment object class car, any other object is taken as background. This is different from the VOC segmentation challenge in which the segmentation result should contain all object classes simultaneously. One of the most challenging aspects of these datasets is the ratio of foreground to background pixels for all categories (Table 2). Moreover, the images in these datasets are not taken in a controlled environment and include severe illumination and occlusion. We compare the proposed approach to the baselines on the 6 object categories in Figure 4. As illustrated, the proposed approach significantly outperforms the baselines on this dataset. Listings of state of the art results are available from [32], [33]. Note that the main contribution of this paper is a general learning method for setting parameters. It could be used in conjunction with other segmentation methods that achieve excellent results on these datasets.

Moreover, the results of "Unary" in most cases except for "aeroplane" class is superior to "Hamming", which

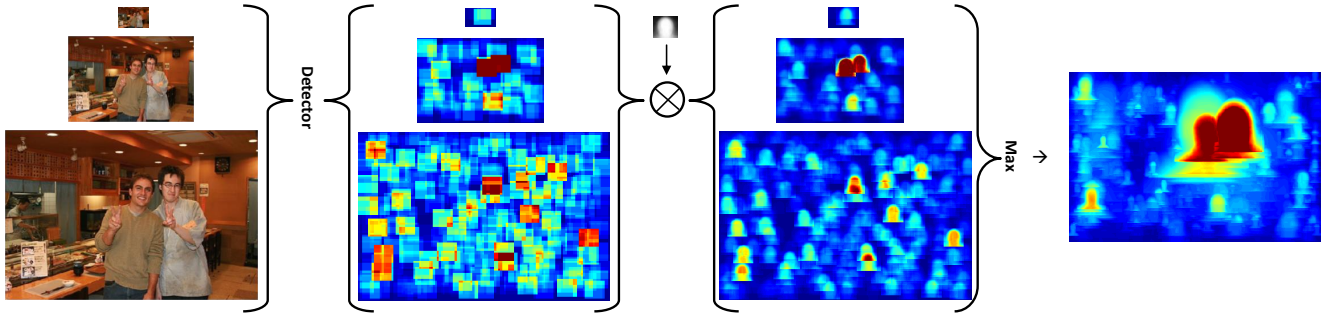


Fig. 3. The process of computing top-down features for the head part. Instead of showing the center of the detected parts we depict the bounding box for visualization purposes in the second stage.

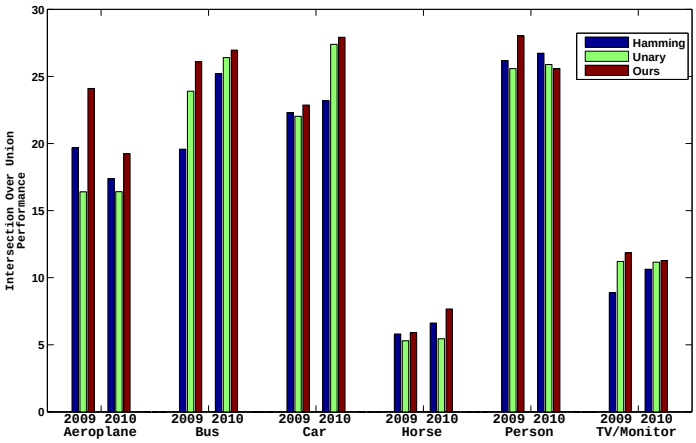


Fig. 4. Intersection over union performance (%) comparison on VOC 2009 and 2010 datasets

Fig. 5. Segmentation for person category. Optimizing adjusted Hamming loss ("Hamming") against our proposed method. a) input image, b) segmentation considering adjusted Hamming loss ("Hamming"), c) our proposed method employing intersection over union. Intersection over union provides more true positives by possibly creating some false positives. Adjusted Hamming loss decreases false positive by sacrificing some true positives.

suggests optimizing against the right performance measure is more important than smoothing the assignments in this dataset.

We compare the effect of optimizing against adjusted Hamming loss versus intersection over union in Fig. 5. Adjusted Hamming loss tends to return fewer false positives, but with the cost of missing many true positives. In fact, it often marks all pixels as background, while intersection over union actually produces segmentations. That is because when the entire image is labeled as background the adjusted Hamming loss results $1/2$ loss while intersection over union loss results 1.

5.5.3 H3D Dataset

We also compare the results on the H3D dataset [34]. This dataset includes 273 training and 107 testing images along with three types of annotations – keypoint annotations, 3d pose annotation and region annotation. The keypoint annotation includes the location of joints and other keypoints such as eyes, nose, elbows, etc. The 3d pose annotation has been inferred from the keypoints. The region annotation, which we use in this paper, provides detailed annotation of people, such as face, neck, lower and upper cloth, etc. For our experiments we compute the union of all region annotations that are part of a person (bags, occluder and hat are not considered as parts of a person) as foreground and the rest as background. The ratio of background to foreground pixels in this dataset is 3.9, which is significantly lower than the ratio in Pascal VOC datasets. The reason is that all images in H3D dataset include at least some foreground pixels, which is not the case in Pascal VOC datasets. The comparison result in Figure 6 shows that the proposed approach outperforms the baselines significantly on this dataset. We also show some segmentation results on H3D dataset in Figure 10.

5.6 Action Retrieval

The second application that we consider in this paper is action retrieval. The task is to find actions that are similar

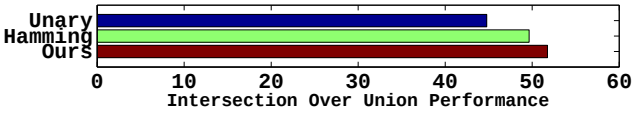


Fig. 6. Intersection over union performance (%) comparison on H3D dataset

to the query action in video frames. In this experiment we are interested in detection and localization of the query action. Action retrieval is an important problem with numerous real-world applications such as multimedia content analysis and surveillance and security systems. In the experiments we explore a surveillance application, automated analysis of nursing home video footage. We would like to find actions of interest – for instance residents falling down, sitting, or standing up. The offline batch processing setting is of interest to clinicians studying the behaviours of nursing home residents. For instance, this setting is useful for gathering data on the circumstances of injurious falls by residents, or mobility measures for residents.

The choice of loss function is arbitrary in our learning framework as long as it remains a function of false positive and false negative counts. A widely used performance measure for retrieval tasks is precision on the first K retrieved elements, termed *precision at K*. This measure represents what we care about when performing retrieval in many applications – one wants to maximize the number of relevant events of interest in a fixed number of retrieved videos. The loss associated with the precision is defined as

$$\Delta_{Prec} = 1 - Precision = \frac{FP}{N_p + FP - FN}, \tag{28}$$

Here, N_p is the number of positive examples (people with ground-truth label equal to the query label). In our action retrieval task, all detections that have the same label as the query action are considered positive and all other detections are negative. So, false positive and false negative counts are computed using Eq. 18 and 19, respectively.

5.6.1 Model

We describe our model for action retrieval in a sequence of video frames. We assume that a set of person locations in each video frame has been provided via a human detection algorithm. The goal is to automatically retrieve the people in a video who perform a query action. We believe there are correlations between the actions of different people in a scene and try to capture these interactions in our action retrieval model.

The model we develop is depicted in Fig. 7. Our model is a Markov Random Field (MRF), where each detection corresponds to a node (site) in the graph (shown in blue). There are three types of edges in the graph, shown in red, green and yellow. Red edges denote the relationship between assigning different labels to each node given the video features describing the corresponding detection. These edges form the unary potentials in our MRF.

The other two edge types model intra-frame and inter-frame correlations between actions. The types of interaction between people in one frame and people in consecutive frames are different. Intra-frame interactions are about which actions are likely to co-occur. On the other hand, inter-frame interactions model the smoothness of people's actions over time. To differentiate the two types of interactions, two groups of pairwise interactions are included in the model, intra-frame interactions and inter-frame interactions, shown in green and yellow, respectively. An edge between two nodes holds a vector of scores corresponding to every possible combination of action labels for its nodes.

Let $\phi(\boldsymbol{x}_i)$ be the feature vector for i^{th} detection and $\mathcal{L}$ be the set of all possible action labels, with M elements. For notational convenience, we encode the action label in a one-of-M format.

Action Appearance Potential θ: The appearance score for the i^{th} node in the graph is formulated as:

$$\theta(\boldsymbol{x}_i, \boldsymbol{y}_i, \overset{u}{\boldsymbol{w}}) = \sum_{k=1}^{M} \overset{u}{\boldsymbol{w}}^T y_{ip} \phi(\boldsymbol{x}_i), \tag{29}$$

Later, in Section 5.6.2 we describe how we compute the appearance features $\phi(\boldsymbol{x})$.

Intra-frame Action Potential ρ: The pairwise action-action scores are only a function of the action labels at two neighbouring nodes with no ordering (symmetric). Under these assumptions, there will be $M(M+1)/2$ parameters. The intra-frame interaction scores between nodes i and j can be written as

$$\rho(\boldsymbol{y}_i, \boldsymbol{y}_j, \overset{p_1}{\boldsymbol{w}}) = \sum_{k=1}^{M} \sum_{l=k}^{M} \overset{p_1}{w}_{k+(l-k)M} y_{ik} y_{jl} \tag{30}$$

Essentially, $\overset{p_1}{\boldsymbol{w}}$ parameters encode which actions are likely to appear together in a frame.

Inter-frame Action Potential μ: Similarly, inter-frame interaction scores can be formulated as

$$\mu(\boldsymbol{y}_i, \boldsymbol{y}_j, \overset{p_2}{\boldsymbol{w}}) = \sum_{k=1}^{M} \sum_{l=1}^{M} \overset{p_2}{w}_{k+lM} y_{ip} y_{jq} \tag{31}$$

with a different set of parameters $\boldsymbol{w}_\mu$ scoring pairs of action labels in consecutive video frames. Note that the transition between actions of a person is not symmetric (walking to falling vs. falling to walking), which results in M^2 parameters for inter-frame potentials in our model.

The overall model score aggregates these cues over a

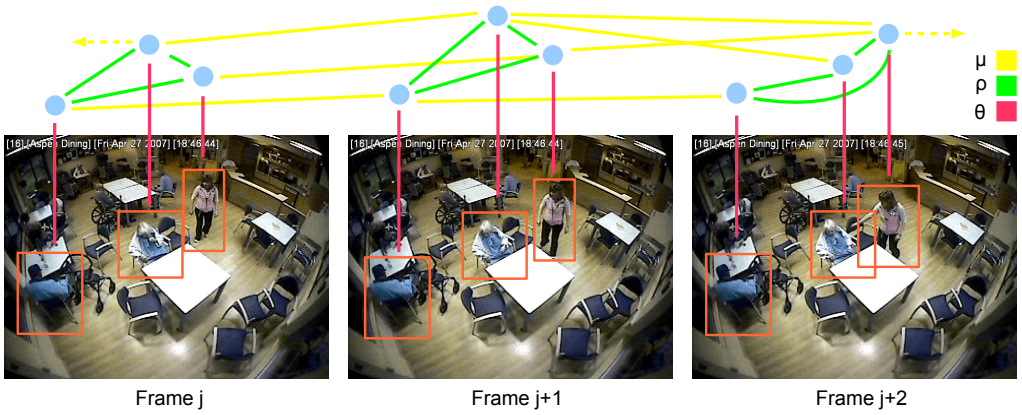


Fig. 7. Illustration of our model. A Markov random field with unary (red), intra-frame (green) and inter-frame (yellow) connections is used.

video sequence, defined as

$$S(\boldsymbol{X},\boldsymbol{Y},\boldsymbol{w}) = \sum_{i=1}^{N}\theta(\boldsymbol{x}_i,\boldsymbol{y}_i,\overset{u}{\boldsymbol{w}})+ \sum_{i=1}^{N}\sum_{j\in\overset{1}{\mathcal{N}}_i}\rho(\boldsymbol{y}_i,\boldsymbol{y}_j,\overset{p_1}{\boldsymbol{w}}) + \sum_{i=1}^{N}\sum_{j\in\overset{2}{\mathcal{N}}_i}\mu(\boldsymbol{y}_i,\boldsymbol{y}_j,\overset{p_2}{\boldsymbol{w}}), \quad (32)$$

where N is the number of nodes in the model (number of detections), $\overset{1}{\mathcal{N}}$ and $\overset{2}{\mathcal{N}}$ are the set of pairs of neighbouring nodes in intra-frame and inter-frame connections, respectively.

5.6.2 Person Detection and Description

We implement a simple method for person detection that proves to be reasonably effective for our dataset. We extract moving regions from the videos using the OpenCV implementation of the standard Gaussian Mixture Model (GMM) [35]. Moving regions with area less than a threshold (500 pixels in our experiments) are deemed unreliable and therefore ignored. In the training set we manually label the output of the detection process from the set of possible actions, which includes the "unknown" action to label the false positives. At test time, we detect people using the same process, extract their features and then recognize their actions.

In our surveillance dataset, widely used features such as optical flow or HOG [36] are typically not reliable due to low video quality. Instead, we use the local spatio-temporal (LST) descriptor [37], which has been shown to be reliable for low spatial and temporal resolution videos. The feature descriptor is computed as follows. We first divide the bounding box of a detected person into N blocks. In the experiments we use a 10×10 grid to obtain 100 blocks for each detection. Foreground pixels are detected using background subtraction. Each foreground pixel is classified as either static or moving by frame differencing. Each block is represented as a vector composed of two components: $\mathbf{u} = [u_1,\ldots,u_t,\ldots,u_\tau]$ and $\mathbf{v} = [v_1,\ldots,v_t,\ldots,v_\tau]$, where u_t and v_t are the percentage of static and moving foreground pixels at time t respectively. τ is the temporal extent used to represent each moving person, which has been set to 5 frames in our experiments.

TABLE 3
Number of detected people in the training and test sets for each action

	unknown	walk	stand	sit	bend	fall
Train	626	331	454	291	38	20
Test	877	330	163	199	13	15

5.6.3 Nursing Home Dataset

We have collected a dataset of 13 video clips from a surveillance camera in a nursing home recorded at 3 frames per second and spatial resolution of 640×480 pixels [38]. The size of the clips in the dataset varies from 94 to 234 frames. the action label set includes 6 actions, unknown, walk, stand, sit, bend and fall. We use 7 clips for training and the remaining 6 clips for testing. After running the detector on all video clips, we manually label all detected bounding boxes. These bounding boxes are employed for training and testing. The summary of the number of detections for each action in the dataset is presented in Table 3. Note that the actions are highly imbalanced and there are only a few detected people with fall, bend actions. We choose the action query label from a subset of these actions – walk, stand, sit, bend and fall.

We fix the value of K to N_p in the experiments and compare three approaches based on precision at K retrieved items. The results are shown in Table 4. The proposed approach outperforms the other approaches for all the actions except *bending*, which has the fewest instances in the test set.

We visualize the intra-frame and inter-frame interaction weights in Fig. 8. One interesting observation is the positive intra-frame weight between *bending* and *walking* while looking for the *walking* action. The *bending* action usually happens in the nursing home dataset when a

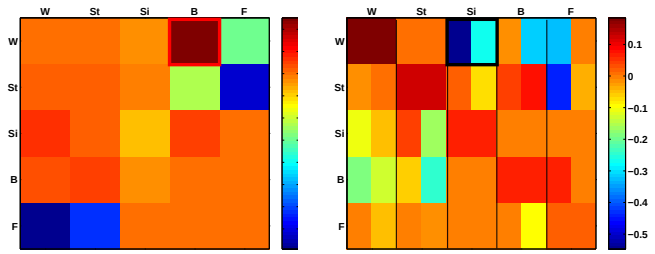


Fig. 8. Visualization for some of the learned intra-frame (left) and inter-frame (right) interactions. Vertical labels are the query actions (walk (W), Stand(St), Sit(Si), Bend(B) and Fall(F)). The inter-frame interactions are asymmetric, which is shown as two weights one from query action to the other actions (left half) and from the other actions to the query action (right half).

TABLE 4
Precision percentage at K $= N_p$

	walk	stand	sit	bend	fall
Hamming	54.1	17.1	36.5	0.0	11.1
Unary	53.5	16.3	38.0	**8.8**	11.1
Our	**55.2**	**17.8**	**38.4**	2.9	**22.2**

nurse is helping an elderly resident who has fallen. In this scenario another nurse is very likely to come to help, who performs the *walking* action. As another example, a person is very unlikely to switch his action from *walking* to *sitting* and vise-versa (Fig 8 right). Also, repeatedly performing the same action over time is likely for all actions except abrupt actions like *falling*.

5.7 Significance Test

As suggested by [39], non-parametric statistical tests such as the Friedman test are more suitable for comparing two or more classifiers over multiple datasets. We follow the approach in [39] (Friedman test + a post-hoc test), and verified that the improved performances w.r.t. the baselines (unary & hamming loss) over all of the datasets in our experiments are significant at $\alpha = 0.05$: the average rank differences between our method and the baselines (1.31 and 1.19) are both larger than the critical difference (0.78).

6 Conclusion

In this paper we developed a general algorithm for addressing learning problems with complex models and complex loss functions, those which are a function of false positive and false negative counts. We replace the original non-decomposable loss function with a piecewise linear approximation, and solve it using a linear programming relaxation of the original quadratic program.

In future work it would be interesting to analyze the quality of these approximations. However, in this work we have provided experimental evidence of their effectiveness. In particular we apply this method to learning an object category segmentation model that contains both unary terms for labeling pixels and pairwise terms on the labels of adjacent superpixels. We show that learning the parameters to this model under an objective directly tied to the performance measure significantly improves performance relative to baseline algorithms on the PASCAL VOC Segmentation Challenges and H3D datasets. Moreover, we proposed a new model for action retrieval that can capture three sources of information: body motion, intra-frame action interaction and inter-frame action interaction. We showed empirically that the proposed approach can significantly improve on two strong baselines, one including our structured model of all actions in a scene, but optimizing decomposable Hamming loss; and the other one optimizing the desired loss function, but without any interaction between different people's actions. Together, these experiments provide evidence that our learning approach can be used to improve the performance of systems using other features and structured models for complex problems.

References

[1] D. Hoiem, A. A. Efros, and M. Hebert, "Closing the loop in scene interpretation," in *CVPR*, 2008.
[2] M. B. Blaschko and C. H. Lampert, "Learning to localize objects with structured output regression," in *ECCV*, 2008.
[3] C. Desai, D. Ramanan, and C. Fowlkes, "Discriminative models for multi-class object layout," in *ICCV*, 2009.
[4] J. Malik, S. Belongie, T. Leung, and J. Shi, "Contour and texture analysis for image segmentation," *IJCV*, vol. 43, pp. 7–27, 2001.
[5] Y. Boykov, O. Veksler, and R. Zabih, "Fast approximate energy minimization via graph cuts," *PAMI*, vol. 23, no. 11, 2001.
[6] M. Szummer, P. Kohli, and D. Hoiem, "Learning crfs using graph cuts," in *ECCV*, 2008.
[7] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The Pascal Visual Object Classes (VOC) Challenge," *IJCV*, vol. 88, no. 2, Jun. 2010.
[8] M. Ranjbar, G. Mori, and Y. Wang, "Optimizing complex loss functions in structured prediction," in *ECCV*, 2010.
[9] B. Taskar, C. Guestrin, and D. Koller, "Max-margin markov networks," in *NIPS*, 2003.
[10] I. Tsochantaridis, T. Joachims, T. Hofmann, and Y. Altun, "Large margin methods for structured and interdependent output variables," *JMLR*, vol. 6, September 2005.
[11] C. H. Teo, A. Smola, S. V. Vishwanathan, and Q. V. Le, "A scalable modular convex solver for regularized risk minimization," in *Proceedings of the 13th ACM SIGKDD*, 2007.
[12] B. Taskar, S. Lacoste-julien, and M. I. Jordan, "Structured prediction via the extragradient method," in *NIPS*, 2005.
[13] S. Shalev-Shwartz, Y. Singer, and N. Srebro, "Pegasos: Primal estimated sub-gradient solver for svm," in *ICML*, 2007.
[14] B. Taskar, V. Chatalbashev, D. Koller, and C. Guestrin, "Learning structured prediction models: a large margin approach," in *ICML*, 2005.
[15] T. Joachims, "A support vector method for multivariate performance measures," in *ICML*, 2005.
[16] Y. Yue, T. Finley, F. Radlinski, and T. Joachims, "A support vector method for optimizing average precision," in *SIGIR*, 2007.
[17] S. Chakrabarti, R. Khanna, U. Sawant, and C. Bhattacharyya, "Structured learning for non-smooth ranking losses," in *Proceeding of the 14th ACM SIGKDD*, 2008.
[18] D. McAllester, T. Hazan, and J. Keshet, "Direct loss minimization for structured prediction," in *NIPS*, 2010.
[19] T. Finley and T. Joachims, "Training structural SVMs when exact inference is intractable," in *ICML*, 2008.

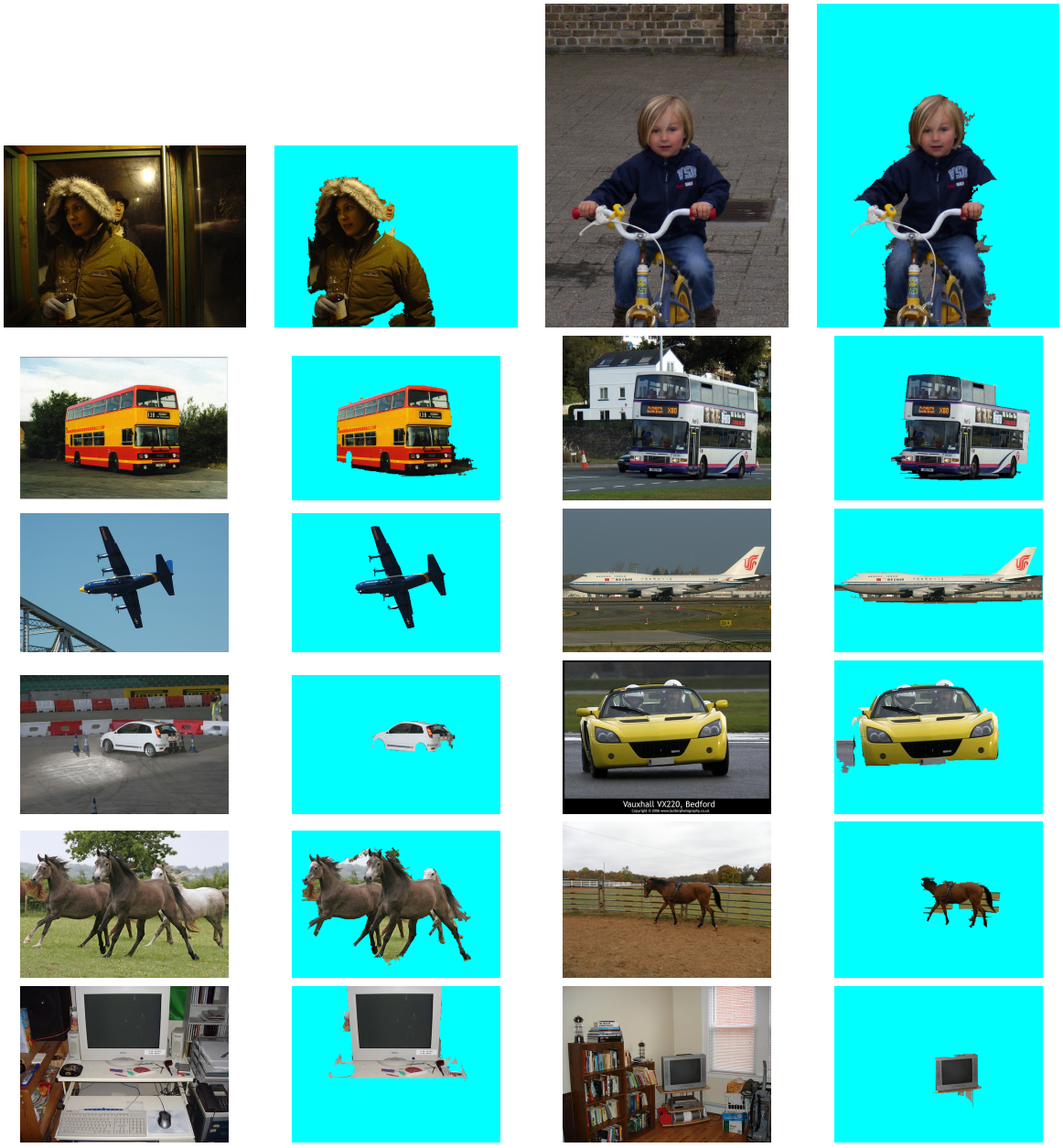


Fig. 9. Some segmentation results on Pascal VOC 2009 dataset. Each row corresponds to one object category.

[20] I. Tsochantaridis, T. Hofmann, T. Joachims, and Y. Altun, "Support vector machine learning for interdependent and structured output spaces," in *ICML*, 2004.
[21] T. Werner, "A linear programming approach to max-sum problem: A review," *IEEE Trans. PAMI*, vol. 29, no. 7, 2007.
[22] N. Komodakis, N. Paragios, and G. Tziritas, "MRF energy minimization and beyond via dual decomposition," *PAMI*, vol. 33, 2011.
[23] Y. Boykov and V. Kolmogorov, "An experimental comparison of min-cut/max-flow algorithms for energy minimization in vision," *PAMI*, vol. 26, 2001.
[24] T. Do and T. Artieres, "Large margin training for hidden markov models with partially observed states," in *ICML*, 2009.
[25] O. Meshi, D. Sontag, T. Jaakkola, and A. Globerson, "Learning efficiently with approximate inference via dual losses," in *ICML*, 2010.
[26] MeshLab, http://meshlab.sourceforge.net/.
[27] M. Garland and P. S. Heckbert, "Surface simplification using quadric error metrics," in *SIGGRAPH*, 1997.
[28] P. F. Felzenszwalb and D. P. Huttenlocher, "Efficient graph-based image segmentation," *IJCV*, vol. 59, no. 2, 2004.
[29] K. E. A. van de Sande, T. Gevers, and C. G. M. Snoek, "Evaluating color descriptors for object and scene recognition," *IEEE Trans. PAMI*, vol. 32, no. 9, 2010.
[30] B. Leibe, A. Leonardis, and B. Schiele, "Combined object categorization and segmentation with an implicit shape model," in *In ECCV Workshop on Statistical Learning in Computer Vision*, 2004.
[31] P. Felzenszwalb, R. Girshick, D. McAllester, and D. Ramanan, "Object detection with discriminatively trained part based models," *IEEE Trans. PAMI*, vol. 32, no. 9, 2009.
[32] M. Everingham, L. Van Gool, C. K. I. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2009 (VOC2009) Results," http://www.pascal-network.org/challenges/VOC/voc2009/workshop/index.html, 2009.
[33] M. Everingham, L. Van Gool, C. K. Williams, J. Winn, and A. Zisserman, "The PASCAL Visual Object Classes Challenge 2010 (VOC2010) Results," http://www.pascal-network.org/challenges/VOC/voc2010/workshop/index.html, 2010.

Fig. 10. Some segmentation results on H3D dataset.

[34] L. Bourdev and J. Malik, "Poselets: Body part detectors trained using 3d human pose annotations," in *ICCV*, 2009.
[35] C. Stauffer and W. E. L. Grimson, "Learning patterns of activity using real-time tracking," *PAMI*, vol. 22, no. 8, pp. 747–757, 2000.
[36] N. Dalal and B. Triggs, "Histograms of oriented gradients for human detection," in *CVPR*, 2005.
[37] C. C. Loy, T. Xiang, and S. Gong, "Modelling activity global temporal dependencies using time delayed probabilistic graphical model," in *ICCV*, 2009.
[38] T. Lan, Y. Wang, G. Mori, and S. Robinovitch, "Retrieving actions in group contexts," in *Int. Work. on Sign Gest. Act.*, 2010.
[39] J. Demsar, "Statistical comparisons of classifiers over multiple data sets," *JMLR*, vol. 7, pp. 1–30, 2006.

Mani Ranjbar is currently a PhD candidate at the School of Computing Science, Simon Fraser University, Canada. He received his M.Sc. in Computer Architecture from Sharif University of Technology, Iran in 2007 and his B.Sc. in Computer Engineering from the same university in 2005. His research interests are in computer vision and machine learning including object detection, segmentation and tracking.

Tian Lan is currently a Ph.D. candidate in the School of Computing Science at Simon Fraser University, Canada. He received his M.Sc. from the same university in 2010, and his B.Eng from Huazhong University of Science and Technology, China in 2008. He has worked as a research intern at Disney Research Pittsburgh in summer 2011. His research interests are in the area of computer vision, with a focus on semantic understanding of human actions and group activities within a scene.

Yang Wang is currently an NSERC postdoctoral fellow at the Department of Computer Science, University of Illinois at Urbana-Champaign. He received his Ph.D. from Simon Fraser University (Canada), his M.Sc. from University of Alberta (Canada), and his B.Sc. from Harbin Institute of Technology (China), all in computer science. He was a research intern at Microsoft Research Cambridge in summer 2006. His research interests lie in high-level recognition problems in computer vision, in particular, human activity recognition, human pose estimation, object/scene recognition, etc.

Steven N. Robinovitch PhD (BAppSc-88, M.S.-90, Ph.D.-95), is a Professor and Canada Research Chair in Injury Prevention and Mobility Biomechanics at Simon Fraser University. His research focuses on improving our understanding of the cause and prevention of fall-related injuries (especially hip fracture) in older adults, through laboratory experiments, mathematical modeling, field studies in residential care facilities, and product design.

Ze-Nian Li is a Professor in the School of Computing Science at Simon Fraser University, British Columbia, Canada. Dr. Li received his undergraduate education in Electrical Engineering from the University of Science and Technology of China, and M.Sc. and Ph.D. degrees in Computer Sciences from the University of Wisconsin-Madison under the supervision of the late Professor Leonard Uhr. His current research interests include computer vision, multimedia, pattern recognition, image processing, and artificial intelligence.

Greg Mori received the Ph.D. degree in Computer Science from the University of California, Berkeley in 2004. He received an Hon. B.Sc. in Computer Science and Mathematics with High Distinction from the University of Toronto in 1999. He is currently an associate professor in the School of Computing Science at Simon Fraser University. Dr. Mori's research interests are in computer vision, and include object recognition, human activity recognition, human body pose estimation.