On Two Methods for Semi-Supervised Structured Prediction

Daniel Munoz J. Andrew Bagnell Martial Hebert

CMU-RI-TR-10-02

January 2010

Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

Abstract

Obtaining labeled data for training classifiers is an expensive task that must be done in any new application. It is yet even more expensive for structured models, such as Conditional Random Fields, where some notion of coherence in the labeled data must be maintained. To address this issue, semi-supervised methods are often applied to reduce the needed number of labeled examples to train adequate models. Previous work in this area have resulted in complex training procedures that do not scale to handle large amounts of examples, features, labels, and interactions that are necessary for vision tasks. In this paper, we present and analyze two novel approaches for semi-supervised training of structured models that can satisfy the above requirements. While we unfortunately do not observe significant benefit from using unlabeled data in our real-world experiments, the simple algorithms we present here may be useful in other applications where the necessary assumptions are satisfied.

Acknowledgements

D. Munoz was supported by a QinetiQ North America Robotics Fellowship. We thank Willow Garage for their Max-Margin Markov Network functional gradient boosting code, V. Kolmogorov for his graph-cut code, and D. Arthur for his k-means code, all of which this work was built upon. We thank N. Vandapel for discussions and improving the presentation.

Contents

1 Introduction

Structured prediction models such as Conditional Random Fields have demonstrated to be strong tools for a variety of vision tasks. However, the ability of these sophisticated models to capture dependencies among the data comes at the price of a more sophisticated training process. Because the predictions are structured, the labeled data needed for training these models must also encode this structure, *e.g.* it is not useful to train a CRF from a random sample of pixels from an image. This expensive labeling approach is not viable in the long term for applications to many different domains.

This problem motivates a semi-supervised approach where unlabeled examples, in addition to a handful of labeled examples, are incorporated into the learning process, as illustrated in Figure 1. The basic idea is that the unlabeled examples provide extra information to the learning algorithm which is not solely captured from the labeled examples and thus leads to an overall better model.

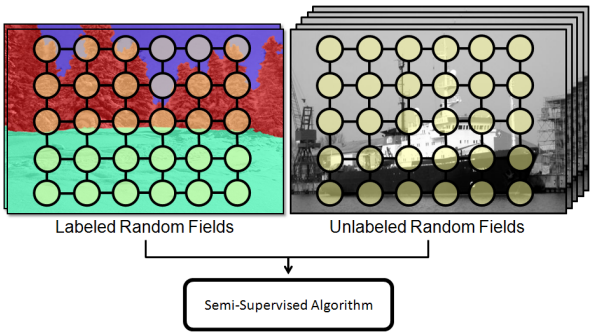
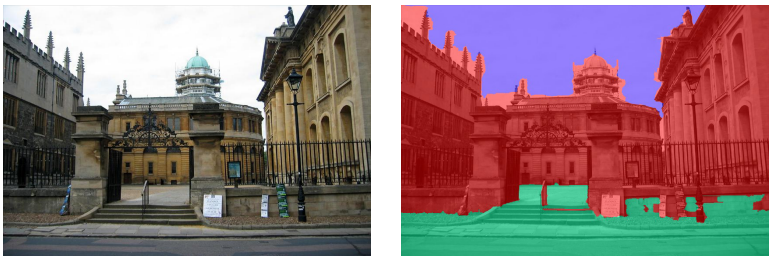


Figure 1: Semi-supervised structured prediction: we wish to incorporate information from many unlabeled random fields (right) in conjunction with a handful of labeled random fields (left) during the learning process.

Many semi-supervised methods impose some notion of smoothness or regularization in the model. For example, in the manifold regularization framework [3], a penalty term is added which practically enforces that similar training examples in feature space should produce similar scores; we further discuss this in Section 3.1. And in the entropy regularization framework [8], a different penalty term is added to train the model such that the unlabeled examples strongly prefer certain labels while the labeled examples are still consistent with their given labels. There has been a lot of recent work on semi-supervised approaches in both the learning [20, 6, 18, 27] and vision [24, 17, 23] communities that are closely related to/inspired from these frameworks; however, most approaches have focused on the unstructured case. Due to the more complex training procedures of structured models, even in the standard supervised case, it is unsurprising that there has been relatively much less attention in the semi-supervised setting [16, 25, 1, 19, 5]. Perhaps due to the large input space for visual analysis, *e.g.* there are easily thousands of pixels in only one example training image, only one of these approaches [16] has analyzed semi-supervised structured models with visual data.

This work investigates the problem of effectively training these structured predic-



(a) (b)

Figure 2: Example Geometric Context classification: (a) Input image (b) The supporting ground (green), vertical structures (red) and Sky (blue) are classified.

tion models in a semi-supervised manner for use in real-world vision tasks. Specifically, we require the optimization procedure to scale with large datasets and random fields which are almost certainly needed for vision tasks. In our experiments we examine the application of Geometric Context surface estimation [9], as illustrated in Figure 2, for evaluation purposes. First we investigate the use of a common semi-supervised framework: manifold reglarization [3]. While previous work [1] has also examined this approach, the authors' proposed optimization procedure involves solving a very large matrix inversion and a convex program, both of which are in practice too computationally expensive to solve in most useful scenarios. Instead, we present a much simpler gradient-based optimization procedure that scales in practice for use in real-world datasets. Unfortunately, while we successfully demonstrate the approach on synthetic datasets, we find in our experiments that this approach often performs *worse* than a solely supervised structured model. In response to this, we present and analyze a second gradient-based technique for using unlabeled random fields. Unfortunately, we do not find any significant benefit with using unlabeled data during the training procedure. While we observe negative results in our experiments, the techniques presented here are applicable to other max-margin structured prediction tasks (*e.g.* sequence labeling) and/or may be useful when the data satisfy the necessary assumptions.

The paper is structured as follows. We first review and introduce notation regarding random fields. In Section 3, we present our gradient-based procedure for semi-supervised learning with manifold regularization and analyze its performance. Finally, in Section 4, we describe and analyze the second procedure for semi-supervised structured prediction.

2 Background

We start with the definition of a Conditional Random Field (CRF) from Lafferty *et al.*[14]. We consider the joint-conditional distribution of our random variables $Y_i \in \mathcal{L}$, *e.g.* the label of a pixel in the image, conditioned on the data X, *e.g.* the features we extract from the image. This distribution is defined by an undirected graph with nodes Y and edges indicating interactions between variables. Let C be the set of cliques in the graph, then each clique $c \in C$ in the graph is associated with a potential function $\phi_c(y_c)$

that measures compatibility of an assignment y_c, *e.g.* the labels of a subset of pixels. To simplify notations, it is implicit that the potential functions ϕ_c are also a function of all the data x. The distribution is then defined by $\log P(y|x) = \Phi(y,x) - \log Z$, where

$$\Phi(y,x) = \sum_{c \in C} \phi_c(y_c), \tag{1}$$

and $Z = \sum_{y' \in \mathcal{Y}} \prod_{c \in C} \exp(\phi_c(y'_c))$ is the normalizer and $\mathcal{Y}$ is the exponential space of possible label configurations.

In this work, we follow the approach of [22, 21] and learn a non-parametric model that is some general function ψ of the features:

$$\phi_c(y_c) \propto \psi(f_c(y_c)), \tag{2}$$

where $f_c \in \mathbb{R}^d$ can intuitively be thought of as features extracted from clique c that describe the assignment y_c; it is also implicit that f_c has access to all the data x. As will be further discussed in the next section, ψ is obtained through a boosting style procedure and therefore the final form of ψ is a weighted combination of simpler functions h: $\psi = \sum_t \eta_t h_t$. Also, we focus on max-margin learning of the model, instead of *maximum a-posteriori* (MAP) learning. Hence these models are referred to as Max-Margin Markov Networks (M^3Ns) [26].

Given labeled training data $(x, \hat{y})$, we learn ψ by minimizing the convex structured-margin loss represented by the functional

$$L[\psi] = \max_{y \in \mathcal{Y}} (\Phi(y,x) + M(y,\hat{y})) - \Phi(\hat{y},x), \tag{3}$$

where $M(y,\hat{y})$ is the structured-margin term analogous to the scalar-margin in Support Vector Machines (SVMs); in our experiments, we use the Hamming loss $M(y,\hat{y}) = \sum_i I[y_i \neq \hat{y}_i]$, where I is the indicator function. Simply, the objective is to find the function ψ that minimizes the difference of the overall score computed with the ground truth labels (second term) and the best overall score compute from any possible labeling by some margin of score (first term). Note that as with most boosting algorithms, we do not have any explicit regularization of ψ over the labeled training data. As is typically done, we instead regularize by having ψ composed of functions h_t with small complexity such as trees with small depth or neural networks with small number of hidden nodes/layers, *etc*.

Labeling random fields is an expensive task in most applications. The remainder of this paper analyzes two approaches for incorporating both labeled cliques C_L and unlabeled cliques C_U during the learning process.

3 Semi-supervision by regularization

As previously discussed, semi-supervised methods can be viewed as imposing a form of regularization over the entire dataset (both labeled and unlabeled examples) during the optimization procedure while still achieving good loss over the labeled examples. That is, the objective is to minimize a functional $\mathcal{O}$:

$$\psi^* = \arg\min_{\psi} \mathcal{O}[\psi] = \arg\min_{\psi} L[\psi] + \lambda \Omega[\psi], \tag{4}$$

where L is the loss over the labeled examples and Ω regularizes ψ over all examples with λ controlling the influence of this term. While any regularization function could be applied here, we examine the manifold regularization framework for semi-supervised structured prediction as Altun *et al.*[1] demonstrated some benefit with this approach.

3.1 Manifold regularization

One method to incorporate unlabeled data into the learning process is with manifold regularization from Belkin *et al.*[3]. Informally, the motivation behind this framework is to regularize the model over the intrinsic distribution of the data. With the assumptions that the data is supported on a lower-dimensional manifold and that the labels change smoothly, this regularization can then be enforced with the graph Laplacian. The Laplacian is defined over all the labeled and unlabeled data to enforce smoothness in how the labels change among neighboring examples in feature space. The Laplacian can augment any loss function, *e.g.* hinge-loss (SVMs), enabling a flexible framework for semi-supervised learning.

Altun *et al.*[1] give a structured version of this regularization term, which we refer to as the Structured Graph Laplacian (SGL), that extends to structured loss functions as with M^{3}Ns:

$$\Omega[\psi] = \sum_{y \in \mathcal{Y}} \sum_{c \in S} \sum_{y' \in \mathcal{Y}} \sum_{c' \in S} W_{c,c'}(y_c, y'_{c'})(\phi_c(y_c) - \phi_{c'}(y'_{c'}))^2, \tag{5}$$

where W is a similarity matrix such as $W_{c,c'}(y_c, y'_{c'}) = \exp(\frac{-1}{2\sigma^2}||f_c(y_c) - f_{c'}(y'_{c'})||^2)$ and $S = C_L \cup C_U$. The main drawback of the optimization procedure presented in [1] is the requirement to invert a $\gamma \times \gamma$ matrix, where γ is the number of labeled and unlabeled cliques times the number of configurations per clique. For example, γ can be greater than $1,000,000$ in the 7-label Geometric Context application. Instead, in the remainder of this section we will present an alternate optimization procedure using gradient-based methods. Because no matrix inverters or convex program solvers are needed, this approach scales for models used in vision tasks with large datasets with large feature dimensions and many labels. We first consider the gradient of the second term, the SGL.

3.2 Regularizer functional gradient

In general, Equation 5 contains an exponential number of terms due to the number of cliques and possible label configurations; however, this sum cannot be that expressive in practice. Since it is NP-complete to perform inference with arbitrary potentials, let alone learning them, we must consider simpler potential functions. Typically one option is to consider potentials over small maximal clique sizes and/or graphs with no cycles. Another recent option from Kohli *et al.*[10] allows for efficient approximate inference over large cliques, though using less expressive potentials that follow the Pott's/associative model:

$$\phi_c(y_c) = \begin{cases} \psi(f_c(a)) \geq 0 & , y_i = a, \forall y_i \in y_c, a \in \mathcal{L} \\ 0 & , \text{otherwise.} \end{cases} \tag{6}$$

That is, the potential can be non-zero only if all the labels in the clique are the same. Networks with these potentials are called Associative Markov Networks (AMNs). Because these type of potentials have demonstrated to be useful in many vision tasks [10, 13, 2, 21], we focus on this case. An alternative way, to express the potential is $\phi_c(y_c) = \sum_{a \in \mathcal{L}} \xi_a(y_c)\psi(f_c(a))$, where $\xi_a(y_c) = \prod_{i \in c} I[y_i = a]$; this form will be useful in Section 3.3.

Firstly, we can simplify the Equation 5 by noting that the vast majority of the label configurations will have zero potential, and we can therefore go from enumerating over all label configurations to enumerating over all labels $a \in \mathcal{L}$ in the training set. Secondly, because the SGL is enforcing that neighboring cliques in feature space have similar score, it is reasonable to ignore pairs where the cliques are assigned different labels, *i.e.* not enforcing that the features that describe a tree should produce the same score as the features that describe a building. Thirdly, it is also reasonable to ignore terms where cliques have different number of nodes, such as ignoring comparisons of node potentials with edge potentials. In the high-order case, a possible strategy would be to compare only the same potentials from cliques/segments generated from the same clustering method, such as segments from mean-shift that use the same bandwidth. Finally, as the similarity function W decays as a function of distance, it is common [3] to only consider a clique feature's k-nearest neighbors, which can be efficiently precomputed in practice, instead of considering a term between all points in feature space.

Using these justifications, we can greatly simplify the form of the SGL as

$$\Omega[\psi] = \sum_{c \in S} \sum_{a \in \mathcal{L}} \sum_{c' \in \mathcal{N}_{c,a}} W_{c,c'}(a,a)(\phi_c(a) - \phi_{c'}(a))^2, \tag{7}$$

where $\mathcal{N}_{c,a}$ are the neighbors of $f_c(a)$. Because we will be using gradient-based methods, we can sequentially evaluate each term in the SGL in practice and do not have to represent the entire term in matrix form. Specifically we use tools from [7, 22], to compute the negative functional gradient $-\nabla_f$ of the SGL:

$$-\nabla_f \Omega[\psi] \propto \sum_{c \in S} \sum_{a \in \mathcal{L}} \omega_{c,a} \delta_{f_c(a)}, \tag{8}$$

where $\delta_{f_c(a)}$ is the Direc-delta function centered at location $f_c(a)$ in function space, and $\omega_{c,a}$ is the SGL functional gradient residual for clique c when labeled a:

$$\omega_{c,a} = \sum_{c' \in \mathcal{N}_{c,a}} W_{c,c'}(a,a)(\phi_{c'}(a) - \phi_c(a)). \tag{9}$$

As will be discussed shortly, we will fit a function at each feature location with response equal to its functional gradient residual. Consider a term $\omega_{c,a}$ with one neighbor c' and the case when $\phi_c(a) > \phi_{c'}(a)$. This implies the residual is negative $\omega_{c,a} < 0$ and that we will update with a function that has negative response evaluated at $f_c(a)$. This is a sensible update as we are decreasing $\phi_c(a)$ to have a value closer to $\phi_{c'}(a)$, *i.e.* we are making the function ψ smoother with respect to its neighbors in features space. Similarly with the term $\omega_{c',a} > 0$, we will update with a function that increases at evaluation $\phi_{c'}(a)$.

3.3 Loss functional gradient

We can similarly take a negative functional subgradient (since the $\max$ operator is non-differentiable) of the first term $L[\psi]$ in Equation 4. As further discussed in Munoz *et al.*[21],

$$-\nabla_f L[\psi] = \sum_{c \in C_L} \sum_{a \in \mathcal{L}} \xi_a(\hat{y}_c)\delta_{f_c(a)} - \xi_a(y_c^*)\delta_{f_c(a)}, \tag{10}$$

where $y^* = \arg\max_{y \in \mathcal{Y}} \Phi(y, x) + M(y, \hat{y})$ results from the inference procedure; in our experiments we use graph-cuts [12, 4]. Consider the case when $\hat{y}_c = a$ and $y_c^* = b$. This also results in a sensible update as we will update with a function that has positive response when evaluating with the true labels $f_c(a)$ and negative response when evaluating with the incorrectly inferred labels $f_c(b)$. We can also use Robust Potts [11] potentials that are not as strict and allow partial inhomogeneous cliques with positive score; we refer to [21] for a more in-depth discussion.

3.4 Putting it all together

We are now ready to compute the complete negative functional gradient of our original objective:

$$-\nabla_f \mathcal{O}[\psi] = -\nabla_f L[\psi] - \lambda \nabla_f R[\psi]. \tag{11}$$

By collecting the residuals with respect to the labeled and unlabeled cliques $c \in C_L$, $c' \in C_U$, respectively, from Equations 8 and 10 we have:

$$-\nabla_f \mathcal{O}[\psi] = \sum_{a \in \mathcal{L}} \Big[\sum_{c \in C_L} \alpha_{c,a}\delta_{f_c(a)} + \sum_{c' \in C_U} \beta_{c',a}\delta_{f_{c'}(a)} \Big], \tag{12}$$

where $\alpha_{c,a}$ and $\beta_{c',a}$ are the functional gradient residuals for the labeled and unlabeled cliques, respectively:

$$\alpha_{c,a} = \xi_a(\hat{y}_c) - \xi_a(y_c^*) + \lambda\omega_{c,a}, \tag{13}$$

$$\beta_{c',a} = \lambda\omega_{c',a}. \tag{14}$$

With these definitions, we can conclude the learning procedure by using functional gradient boosting tools from [22, 7]. At each iteration t, we take a step η_t in the direction of the negative functional gradient. Practically, we move by projecting this functional gradient onto the space of candidate functions $\mathcal{H}$ by using the function $h_t^* \in \mathcal{H}$ that best maximizes its inner product with the negative functional gradient:

$$\begin{aligned} h_t^* &= \arg\max_{h_t \in \mathcal{H}} \langle h_t, -\nabla_f \mathcal{O}[\psi] \rangle \\ &= \arg\max_{h_t \in \mathcal{H}} \sum_{\substack{a \in \mathcal{L} \\ c \in C_L}} \alpha_{c,a} h_t(f_c(a)) + \sum_{\substack{a \in \mathcal{L} \\ c' \in C_U}} \beta_{c',a} h_t(f_{c'}(a)). \end{aligned}$$

Simply, this expression suggests that we should update ψ with a function that has signed responses proportional to the residuals α, β at each feature location. Practically, this suggests to create a labeled training set with non-zero target values α, β to

train a regressor; in our experiments we use OpenCV's regression trees constrained in the range $[-1, 1]$. Finally, to ensure the positivity constraints $\psi \geq 0$ of our potentials, we use exponentiated functional gradient descent [22]. Simply, this means that we evaluate our potentials as $\psi(f_c(a)) = \exp(\sum \eta_t h_t(f_c(a)))$. We refer to [22] for more in-depth discussion. The entire training procedure is described in Algorithm 1.

Algorithm 1 Semi-supervised AMNs with manifold regularization

Inputs: Set of training labels: $\mathcal{L}$, Labeled cliques: C_L, Unlabeled cliques: C_U, Step size: η_t, Semi-supervised regularization: λ, Number of iterations: T
Output: AMN model ψ
$\psi = 1$
for $t = 1 \dots T$ **do**
  $y^* = \arg\max_{y \in \mathcal{Y}} \Phi(x, y) + M(y, \widehat{y})$
  Initialize training set $\mathcal{D} = \emptyset$
  for $a \in \mathcal{L}$ **do**
    // Compute residuals over labeled cliques
    for $c \in C_L$ **do**
      Compute residual $\alpha_{c,a}$ (Equation 13)
      if $\alpha_{c,a} \neq 0$ **then**
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(f_c(a), \alpha_{c,a})\}$
      end if
    end for
    // Compute residuals over unlabeled cliques
    for $c' \in C_U$ **do**
      Compute residual $\beta_{c',a}$ (Equation 14)
      if $\beta_{c',a} \neq 0$ **then**
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(f_{c'}(a), \beta_{c',a})\}$
      end if
    end for
  end for
  $h_t \leftarrow$ `trainSupervisedLearner`$(\mathcal{D})$
  $\psi \leftarrow \psi \cdot \exp(\eta_t h_t)$
end for
return ψ

3.5 Experimental analysis

3.5.1 Toy examples

To validate the approach we used the popular `2-circles` and `2-moons`, as depicted in Figure 3, synthetic datasets for semi-supervised learning. Here a small portion of samples are labeled such that when passed to a supervised algorithm it would then usually result in a linear decision boundary that does not represent the true distribution of each class. Since both examples follow the manifold assumption, we can use manifold regularization to learn correct accurate models. In Figure 4, we illustrate the conver-

(a) (b)

Figure 3: Toy synthetic datasets: (a) `2-circles` (b) `2-moons`. Colored points represent the two classes and black points are the unlabeled examples provided during training.

gence of the learning procedures when using a random field with: a) only node potentials, b) and with edge potentials from 2-nearest-neighbors, c) and with high-order clique potentials defined over clusters from mean-shift segmentation. In the training set, we also add an extra edge and high-order clique that contain mixed labels for context. The features of all cliques are their respective centroids. All models eventually converge to the ideal classifier, though we found the structured models naturally make smoother progress due to the influence of the potentials.

3.5.2 Geometric context

We evaluated the approach on the the Geometric Context problem from Hoiem *et al.*[9]. In this problem, we determine whether a pixel belongs in one of three classes: Ground, Vertical (object standing on the ground), Sky. In the following we explain how we cast this as a random field problem. In [9], the authors first group pixels into superpixels to use as the sites to classify. They then perform 15 different segmentations: by increments of 5 segments between 5 and 50, and then by increments of 10 segments until 100. Each segment is composed of a group of superpixels. For each superpixel and segment, they extract 50 and 94 features, respectively, which capture location, shape, color, texture, and perspective. We use these same features. In our random field, a node variable represents each superpixel's class label, and high-order cliques are defined over the superpixels (nodes) contained within one segment. We consider two sets of clique potentials: one shared across all the nodes and one shared across all the high-order cliques. For the high-order cliques, we use the Robust Potts [11] potentials with truncation parameter of $0.1|c|$, *i.e.* we allow 10% of the nodes to disagree with the clique's mode label while still allowing positive potential.

The Geometric Context evaluation dataset consists 250 images. Five-fold evaluation is performed by separating the dataset five times with 200 training images and 50 testing images. When validating parameters (tree-depth, step-size, number of iterations, and manifold regularization parameter), we examined random subsets of the 200 training images from a particular fold.

In the following experiment we compare a supervised model trained with only 5 labeled images with a semi-supervised model trained with 5 labeled and 50 unlabeled

Figure 4: Convergence analysis for toy data sets. From left to right: Rows 1 & 4: Convergence with model trained using only node potentials, Rows 2 & 5: Convergence with model trained using edge and node potentials, Rows 3 & 6: Convergence with model trained with high-order, edge, and node potentials. The last column is the iteration when convergence is achieved for the slowest model; all other iterations were chosen to contrast differences in progress and not when convergence is achieved.

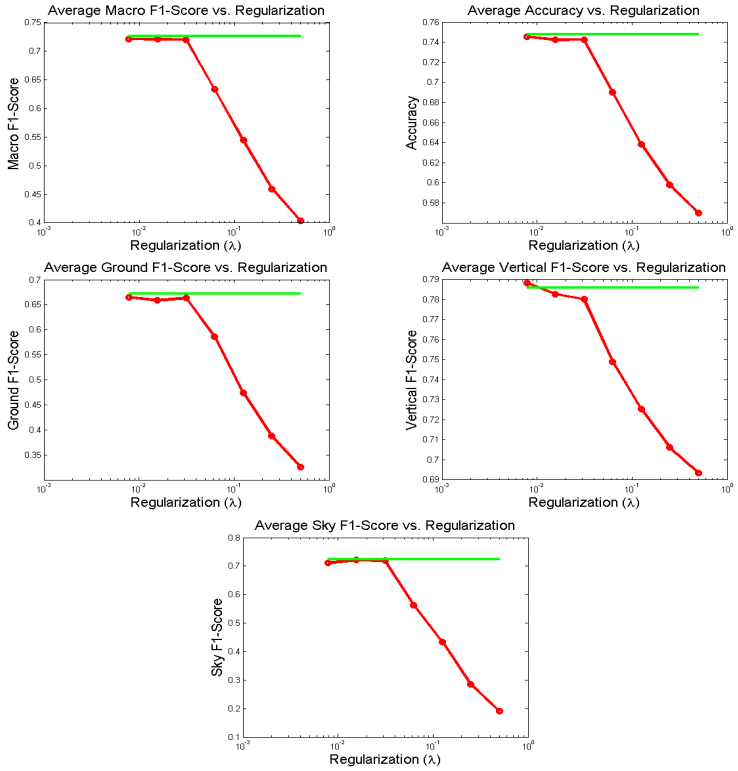


Figure 5: Analysis of Laplacian regularizer averaged over 10 random trials. Green line: statistics from supervised model trained with 5 labeled images (constant). Red line: statistics from semi-supervised model with manifold regularization trained 5 labeled images and 50 unlabeled images. Seven different regularization parameters (λ) are varied (horizontal axis, log scale). Classification is done on one 25-image test set.

images (none were taken from the fold's 50 testing images). For the semi-supervised model, we used a conservative estimation to construct the Laplacian. We construct the graph Laplacian over each clique's 7 nearest neighbor and define $\sigma = 0.08$ in the distance function. Unfortunately, in our validation search we could not find a regularization parameter λ that consistently produced better classification performance in any of the folds. We also observed that the best performance is achieved by setting $\lambda \to 0$, *i.e.* essentially training a supervised model. For the sake evaluation purposes, we picked a fold and trained model models where we varied the regularization λ and compared with the supervised model on the test split. Figure 5 presents this analysis averaging over 10 trials where the labeled and unlabeled images are randomly selected; the Macro F1-score is defined as the average of the three labels' F1-scores. The performance on the test split was consistent with the observation from validation: we obtain no essentially no overall improvement by using unlabeled data.

4 Semi-supervision by gradient modeling

4.1 Motivation

One reason reason why the previous experiments may have shown no benefit with using the unlabeled examples is that the datasets do not follow the manifold assumption. Even though we "cheated" in our post-analysis to verify that our features across labeled and "unlabeled" examples from the same classes were similar to each other, this is not sufficient. Lafferty and Wasserman [15] analyzed this topic and show that a model trained with the Laplacian regularizer in the absence of the manifold assumption does not achieve lower risk than a model trained solely with the labeled examples. In response to this, we explore an alternate method to employ semi-supervised learning. The main idea behind this approach is to take better directions at each step of the boosting procedure. Instead of regularizing over the unlabeled data, we remove this penalty term and instead use the unlabeled data directly at each iteration to help interpret/model the gradient. Figure 6 gives an comparison between the two approaches to help guide our discussion.

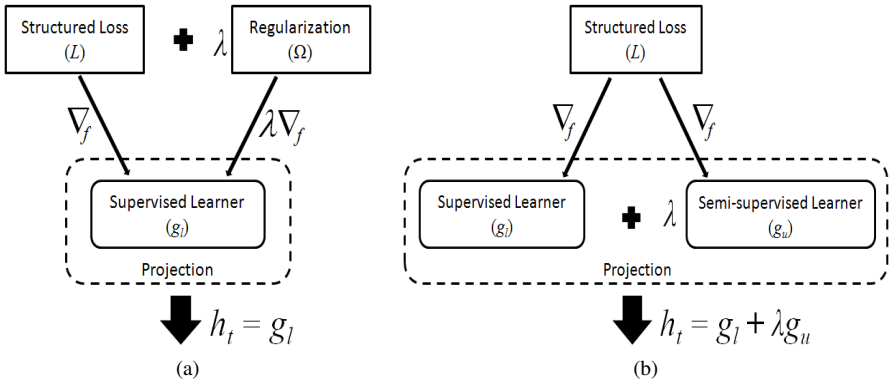


Figure 6: Gradient-based approaches for semi-supervised learning: (a) Compute the direction that also explicitly minimizes the penalty over the unlabeled data. (b) Using the unlabeled data at each iteration to better estimate the gradient direction.

4.2 Approach

With the algorithm from the previous section (Figure 6a), we take a step in a direction that achieves smaller loss and smaller regularization penalty with respect to the unlabeled data. This procedure explicitly tries to model the nature of the data with the penalty term. Instead we propose to move in a direction that achieves smaller loss while implicitly capturing the nature of the data (Figure 6b). Recall the supervised loss (Equation 3). At each step of the boosting procedure we attempt to move in the direction of the functional gradient via the projection procedure of fitting a function according to the residuals centered at various feature locations. With limited number

of labeled examples, this gradient may be sparse and the function approximation may not be able to truly capture the nature of the data. Therefore, at each step we try to better interpret the gradient by using the unlabeled data to fit a better function. Figure 7 demonstrates this idea.

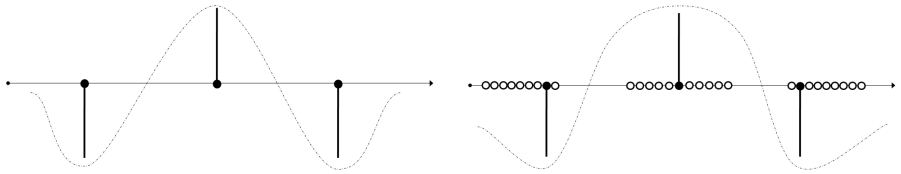


Figure 7: Improving the projection step during functional gradient descent. The horizontal line are the feature locations, the vertical lines are the functional gradient residuals at those locations, filled circles are labeled examples, and empty circles are unlabeled examples. (a) A function is fit to the dataset dictated by the loss' functional gradient (Equation 10) over the labeled examples. (b) We can better interpret the functional gradient by using the unlabeled examples to fit a function that better models the data distribution.

Therefore, instead of fitting a single function during the projection step, we suggest using a combination of two $h = g_l + \lambda g_u$, where g_l is trained over solely the loss residuals and g_u is trained over the loss residuals and all unlabeled data with λ controlling its influence. Just as we could originally train any supervised learning algorithm, we can similarly train any semi-supervised learning algorithm for g_u and treat it as another black-box. In Laffery and Wasserman's analysis they show that a simple clustering procedure using the labeled and unlabeled data for semi-supervised regression achieves lower risk than an estimator that solely uses the labeled data. Given labeled and unlabeled examples, they suggest to create a set of convex clusters over the entire dataset. When given an example at test time, find its nearest cluster and output the average response from the labeled examples contained within that cluster (note that the labels are values since we are doing regression). That is, the unlabeled examples are solely used for creating the clusters. Figure 8 demonstrates this idea. Inspired by this suggestion, we use simple k-means as our clustering tool. Since defining the number of clusters in advance is unclear, we average multiple the responses from multiple clusterings. The final procedure, similar to Algorithm 1, is described in Algorithm 2.

4.3 Experiments

We again analyize the Geometric Context dataset. Unlike with manifold regularization, we observed much more stability when tuning $\lambda = 0.2$ with different values over different trials. That is, we did not observe rapid worsening performance when increasing λ as seen with manifold regularization. In the following experiment we ran the fully 5-fold evaluation and compare a supervised model trained with only 5 labeled images and a semi-supervised model trained with 5 labeled and 50 unlabeled images. The semi-supervised classifier g_u averaged responses from k-mean clusterings with

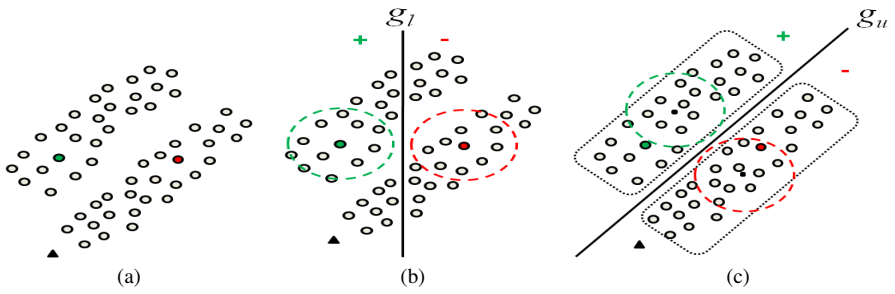


Figure 8: Semi-supervised classification by clustering. (a) Training set consists of 2 labeled points (red, green circles) and many unlabeled points (empty circles). We wish to classify the triangle. (b) Ignoring the labeled data, the supervised algorithm (g_l) results in classifying the test point green. (c) The semi-supervised clustering algorithm (g_u) clusters of all training points; the test point is closer to the cluster labeling it red.

Algorithm 2 Semi-supervised AMNs with gradient modeling

Inputs: Set of training labels: $\mathcal{L}$, Labeled cliques: C_L, Unlabeled cliques: C_U, Step size: η_t, Semi-supervised regularization: λ, Number of iterations: T
Output: AMN model: ψ
$\psi = 1$
for $t = 1 \ldots T$ **do**
  $y^* = \arg\max_{y \in \mathcal{Y}} \Phi(x, y) + M(y, \widehat{y})$
  Initialize training set $\mathcal{D} = \emptyset$
  for $a \in \mathcal{L}$ **do**
    for $c \in C_L$ **do**
      Compute residual $\rho_{c,a} = \xi_a(\hat{y}_c) - \xi_a(y_c^*)$
      if $\rho_{c,a} \neq 0$ **then**
        $\mathcal{D} \leftarrow \mathcal{D} \cup \{(f_c(a), \rho_{c,a})\}$
      end if
    end for
  end for
  $g_l \leftarrow \texttt{trainSupervisedLearner}(\mathcal{D})$
  $g_u \leftarrow \texttt{trainSemiSupervisedLearner}(\mathcal{D}, C_U)$
  $h_t \leftarrow g_l + \lambda g_u$
  $\psi \leftarrow \psi \cdot \exp(\eta_t h_t)$
end for
return ψ

$k = \{6, 9, 12, 15\}$. Figure 9 presents this analysis over 10 trials where the labeled and unlabeled images are randomly selected. Unfortunately, we again do not see positive benefit from incorporating unlabeled examples during the learning process. One reason there may be no improvement is that the clustering classifier is a poor modeler of the gradient. It remains future work to analyze other semi-supervised algorithms.

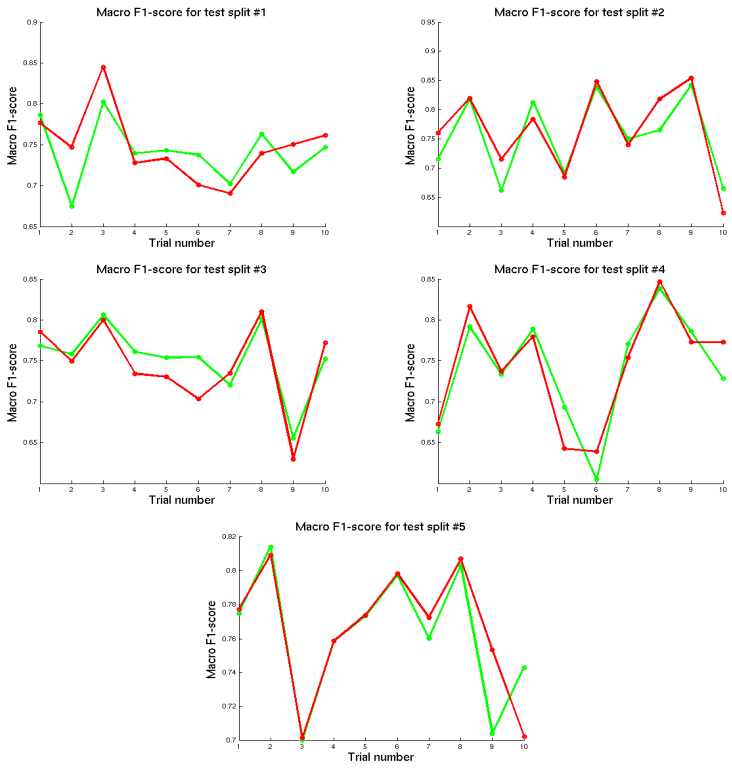


Figure 9: Analysis of semi-supervised gradient estimation from 5-fold evaluation. Green line: macro F1-score from supervised model trained with 5 labeled images. Red line: macro F1-score semi-supervised model with clustering procedure (see text) trained 5 labeled images and 180 unlabeled images. Each fold randomly chooses the training set 10 times.

5 Conclusion

In this paper we presented two functional gradient-based procedures for semi-supervised structured prediction. Specifically, we first showed how manifold regularization for structured models can be efficiently optimized instead of using convex program solvers. The second method uses black-box semi-supervised algorithms to better model the gradient. Both these approaches easily scale to handle large amounts of examples, features, labels, and interactions that are necessary for visual analysis. Unfortunately, we observed in our analysis of Geometric Context [9] that both semi-supervised approaches do not improve classification performance over solely supervised models. Future work includes evaluating these techniques on other datasets where necessary data assumptions may hold true and experimenting with different classifiers that implement the functional gradient.

References

[1] Y. Altun, D. McAllester, and M. Belkin. Maximum margin semi-supervised learning for structured variables. In *NIPS*, 2005. 1, 2, 4

[2] D. Anguelov, B. Taskar, V. Chatalbashev, D. Koller, D. Gupta, G. Heitz, and A. Ng. Discriminative learning of markov random fields for segmentation of 3-d scan data. In *CVPR*, 2005. 5

[3] M. Belkin, P. Niyogi, and V. Sindhwani. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *JMLR*, 7, 2006. 1, 2, 4, 5

[4] Y. Boykov, O. Veksler, and R. Zabih. Fast approximate energy minimization via graph cuts. *T-PAMI*, 23(1), 2001. 6

[5] U. Brefeld and T. Scheffer. Semi-supervised learning for structured output variables. In *ICML*, 2006. 1

[6] K. Chen and S. Wang. Regularized boost for semi-supervised learning. In *NIPS*, 2007. 1

[7] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *Annals of Statistics*, 29, 2001. 5, 6

[8] Y. Grandvalet and Y. Bengio. Semi-supervised learning by entropy minimization. In *NIPS*, 2004. 1

[9] D. Hoiem, A. Efros, and M. Hebert. Recovering surface layout from an image. *IJCV*, 75(1), 2007. 2, 8, 14

[10] P. Kohli, M. P. Kumar, and P. H. Torr. P3 and beyond: Move making algorithms for solving higher order functions. *T-PAMI*, 31(9), 2009. 4, 5

[11] P. Kohli, L. Ladicky, and P. H. Torr. Robust higher order potentials for enforcing label consistency. *IJCV*, 82(3), 2009. 6, 8

[12] V. Kolmogorov and R. Zabih. What energy functions can be minimized via graph cuts? *T-PAMI*, 26(2), 2004. 6

[13] S. Kumar and M. Hebert. Discriminative random fields. *IJCV*, 2006. 5

[14] J. Lafferty, A. McCallum, and F. Pereira. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *ICML*, 2001. 2

[15] J. Lafferty and L. Wasserman. Statistical analysis of semi-supervised regression. In *NIPS*, 2007. 11

[16] C.-H. Lee, S. Wang, F. Jiao, D. Schuurmans, and R. Greiner. Learning to model spatial dependency: Semi-supervised discriminative random fields. In *NIPS*, 2006. 1

[17] C. Leistner, H. Grabner, and H. Bischof. Semi-supervised boosting using visual similarity learning. In *CVPR*, 2008. 1

[18] N. Loeff, D. Forsyth, and D. Ramachandran. Manifoldboost: Stagewise function approximation for fully-, semi- and un-supervised learning. In *ICML*, 2008. 1

[19] M. Mahdaviani and T. Choudhury. Fast and scalable training of semi-supervised crfs with application to activity recognition. In *NIPS*, 2007. 1

[20] P. K. Mallapragada, R. Jin, A. K. Jain, and Y. Liu. Semiboost: Boosting for semi-supervised learning. *T-PAMI*, To Appear. 1

[21] D. Munoz, J. A. Bagnell, N. Vandapel, and M. Hebert. Contextual classification with functional max-margin markov networks. In *CVPR*, 2009. 3, 5, 6

[22] N. Ratliff, D. Silver, and J. A. Bagnell. Learning to search: Functional gradient techniques for imitation learning. *Autonomous Robots*, 27(1), 2009. 3, 5, 6, 7

[23] A. Saffari, H. Grabner, and H. Bischof. Serboost; semi-supervised boosting with expectation regularization. In *ECCV*, 2008. 1

[24] A. Saffari, C. Leistner, and H. Bischof. Regularized multi-class semi-supervised boosting. In *CVPR*, 2009. 1

[25] J. Suzuki, A. Fujino, and H. Isozaki. Semi-supervised structured output learning based on a hybrid generative and discriminative approach. In *ACL*, 2007. 1

[26] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, 2003. 3

[27] H. Valizadegan, R. Jin, and A. K. Jain. Semi-supervised boosting for multi-class classification. In *ECML*, 2008. 1