Multi-utility Learning: Structured-output Learning with Multiple Annotation-specific Loss Functions

Roman Shapovalov **Dmitry Vetrov** **Anton Osokin**
Lomonosov Moscow State University
shapovalov@graphics.cs.msu.ru
vetrovd@yandex.ru anton.osokin@gmail.com

Pushmeet Kohli
Microsoft Research Cambridge
pkohli@microsoft.com

Abstract

Structured-output learning is a challenging problem; particularly so because of the difficulty in obtaining large datasets of fully labelled instances for training. In this paper we try to overcome this difficulty by presenting a multi-utility learning framework for structured prediction that can learn from training instances with different forms of supervision. We propose a unified technique for inferring the loss functions most suitable for quantifying the consistency of solutions with the given weak annotation. We demonstrate the effectiveness of our framework on the challenging semantic image segmentation problem for which a wide variety of annotations can be used. For instance, the popular training datasets for semantic segmentation are composed of images with hard-to-generate full pixel labellings, as well as images with easy-to-obtain weak annotations, such as bounding boxes around objects, or image-level labels that specify which object categories are present in an image. Experimental evaluation shows that the use of annotation-specific loss functions dramatically improves segmentation accuracy compared to the baseline system where only one type of weak annotation is used.

1 Introduction

Training structured-output classifiers is a challenging problem; not only because of the associated computational burden, but also due to difficulties in obtaining the ground-truth labelling for training data: in problems like semantic image segmentation the structured label may comprise thousands

(a) Original image (b) Full (strong) labelling (c) Bnd.-box annotation (d) Object-seed annotation

Figure 1: Types of annotation for an image from the MSRC dataset [15]

of scalars, so annotation of large datasets requires a lot of human effort. In contrast, it is much easier to obtain a weak annotation of an image, i.e. some statistic of the image labelling. This may take various forms: an image-level label that indicates presence or counts the number of pixels of a particular object category like 'sky' or 'water', a set of objects' bounding boxes—rectangles that tightly bound object instances' segmentations, or a set of seeds—the pixels that have to take the specified labels (Fig. 1). More broadly, weakly-supervised learning may be useful in many training problems where the input is obtained by crowdsourcing. For example, some part of a training set for object detection may be of low quality, meaning that the bounding boxes are not tight. In the document tagging problem, low-quality ground truth may miss some tags of the documents. It is preferable to model those biases in the annotation explicitly.

As for semantic segmentation, different types of annotations help not only to overcome logistic difficulties, but also to characterize certain categories better. For example, many object categories (i.e. 'things' in terms of Heitz and Koller [6]) are better described by bounding-box annotations, while the background categories (i.e. 'stuff' [6])—which tend to fill significant parts of an image—by image-level labels.

A number of researchers have recognized the importance of weak annotations for learning semantic segmentation. However, most of these methods only use image-level labels. For example, Vezhnevets et al. [22, 23] use a multi-image probabilistic graphical model to propagate image-level annotations across different training images. In this paper, we present a framework for learning structured classification from the mixture of fully and weakly annotated instances. Our framework can employ different types of weak annotations, even for a single instance.

Our work extends recent research on using latent-variable structural support vector machines (LV-SSVM) for weakly-supervised learning [3, 8, 11] by incorporating *annotation-specific* loss functions, which measure the inconsistency of some labelling predicted by the algorithm with the ground-truth weak annotation. We define those loss functions such that each of them returns an estimate of the expected Hamming loss w.r.t. all possible labellings consistent with the corresponding weak annotation. Due to this definition, the loss functions specific to different annotation types have the same scale. Our framework thus requires only one coefficient, which balances the relative impact of the loss functions for fully labelled and weakly annotated data, since the latter are typically less informative. We empirically show that balancing between these two kinds of loss functions can improve labelling performance.

A number of key technical challenges arise while learning an LV-SSVM model with multiple annotation-specific loss functions. These include solution of the *loss-augmented* and *annotation-consistent* inference problems. The former involves finding the labelling that satisfies the current model and deviates from the annotation the most, while the latter involves finding the best labelling that is consistent with the weak annotation. We show how to solve these optimization problems for various loss functions using efficient optimization algorithms.

Relation to previous work. Our work is most closely related to the work of Kumar et al. [8], who use a sequential method to learn semantic segmentation from different types of annotations. Their method starts by training LV-SSVM with a loss function defined on partial labellings; it performs loss-augmented inference using carefully initialized iterated conditional modes (ICM). Once this model is trained, they infer the partial labellings for weakly-annotated images that are consistent with their bounding-box or image-level annotations. The model is then re-trained by considering those solutions as the true partial labellings for the training instances. Unlike Kumar et al. [8], at the training stage we minimize our annotation-specific loss functions simultaneously. In this regard, our framework does not require neither fully nor partially labelled images, which are essential for the first stage of their algorithm. Furthermore, our loss functions allow us to use powerful graph cut based algorithms for solving the loss-augmented and annotation-consistent inference problems, instead of using an ICM-based inference. Finally, we use different types of weak annotations.

For some of the loss functions we use, the loss-augmented inference problems cannot be decomposed to the individual variables. This relates us to the recent work on supervised learning with non-decomposable loss functions [13, 16]. Pletscher and Kohli [13] use a higher-order loss function that penalizes the difference in the area of the target category between binary segmentations. They show how to use graph cuts for efficient exact loss-augmented inference. Tarlow and Zemel [16]

use message-passing inference in SSVM training with three different higher-order loss functions: PASCAL VOC loss, bounding box fullness loss, and local border convexity loss.

Our contributions:

- we propose an LV-SSVM based multi-utility learning framework, which simultaneously minimizes different annotation-specific loss functions, and a unified technique for establishing loss functions for weak annotation of different types;
- we apply our framework to define the loss functions for training semantic segmentation that are specific to the following weak annotation types and their combinations: image-level labels, bounding boxes, and objects' seeds;
- we propose efficient inference algorithms required for LV-SSVM training with these loss functions.

2 Latent-variable SSVM

2.1 Structured-output learning

Structured-output learning attempts to learn a mapping H from the space of features $\mathcal{X}$ to the space of all possible labellings $\mathcal{Y}$. In what follows, we consider only the mappings that can be expressed as maximization of a discriminant function F that depends linearly on its parameters $\mathbf{w}$:

$$H(\mathbf{x}) = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} F(\mathbf{x},\mathbf{y};\mathbf{w}) = \operatorname*{argmax}_{\mathbf{y}\in\mathcal{Y}} \mathbf{w}^\intercal \boldsymbol{\Psi}(\mathbf{x},\mathbf{y}), \tag{1}$$

where vector function $\boldsymbol{\Psi}(\mathbf{x},\mathbf{y})$ denotes so-called generalized features of instance $\mathbf{x} \in \mathcal{X}$ and labelling $\mathbf{y}$. $\boldsymbol{\Psi}(\mathbf{x},\mathbf{y})$ is defined in a problem-specific way, while the weights $\mathbf{w}$ are learned from the training data. We address a wide class of so-called *labelling problems*, where the structured label is a vector of discrete variables: $\mathcal{Y} = \mathcal{K}^V$, where $\mathcal{K} = \{1,\dots,K\}$. Its length V may vary for individual instances.

The goal of supervised structured-output learning is to obtain the most appropriate weights $\mathbf{w}$ given the set of features and ground-truth labels of training instances: $\{(\mathbf{x}_n,\mathbf{y}_n)\}_{n=1}^N$, $\mathbf{y}_n \in \mathcal{Y}_n$. Here $\mathcal{Y}_n$ is a set of possible labellings compatible with the n-th instance. In this paper we follow the max-margin formulation of structured-output learning (also called structural support vector machine, SSVM) [17, 20, 7]:

$$\min_{\mathbf{w},\boldsymbol{\xi}\geq\mathbf{0}} \ \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{N}\sum_{n=1}^{N}\xi_n, \tag{2}$$

$$\text{s.t.} \quad F(\mathbf{x}_n,\mathbf{y}_n;\mathbf{w}) \geq \max_{\bar{\mathbf{y}}\in\mathcal{Y}_n} \big(F(\mathbf{x}_n,\bar{\mathbf{y}};\mathbf{w}) + \Delta(\bar{\mathbf{y}},\mathbf{y}_n)\big) - \xi_n, \ \ \forall n, \tag{3}$$

where $\Delta(\bar{\mathbf{y}},\mathbf{y}_n)$ is the loss of some labelling $\bar{\mathbf{y}} = \{\bar{y}_i\}_{i=1}^V$ with respect to the ground truth labelling $\mathbf{y}_n = \{y_i^n\}_{i=1}^{V_n}$. Let c_i^n be some cost associated with the i-th variable in the labelling of the n-th instance. The commonly used loss function is the weighted Hamming distance:

$$\Delta(\bar{\mathbf{y}},\mathbf{y}_n) = \sum_{i\in\mathcal{V}_n} c_i^n[\bar{y}_i \neq y_i^n],^{1} \tag{4}$$

This loss function is decomposable w.r.t. the individual variables. It often implies that loss-augmented inference, i.e. maximization in (3), is no more difficult than the maximization of discriminant function $F(\mathbf{x},\mathbf{y};\mathbf{w})$. In some cases it is possible to use higher-order loss functions that cannot be decomposed w.r.t. the individual variables [13, 16, 4].

Problem (2)–(3) is convex and can be solved by the cutting-plane method [20, 7]. This method replaces the constraint (3) with a bunch of linear constraints and then iteratively approximates the feasible polytope by adding the most violated constraint. Such constraint is determined in each iteration by running the loss-augmented inference in (3).

[1] We use the Iverson bracket notation: $[e] = 1$ if the logical expression e is true, and $[e] = 0$ otherwise

2.2 Learning with weak annotations

Consider the case when in addition to N fully-labelled objects, train set contains M weakly-annotated ones: $\{(\mathbf{x}_m, \mathbf{z}_m)\}_{m=N+1}^{N+M}$. Hereinafter we assume that the weak annotation $\mathbf{z}_m$ defines a subset of full labellings $\mathcal{L}(\mathbf{z}_m) \subset \mathcal{Y}$ that are consistent with it, and thus $\mathbf{z}_m$ is less informative than an individual full labelling $\mathbf{y}_m$. Examples of such weak annotations for the image segmentation problem are (1) bounding boxes of the segments of a given label; (2) a value of some global statistic (area, average intensity, number of connected components etc.) for the segments of a given label; (3) subsets of superpixels that belong to a given label (seeds).

We now generalize the standard SSVM formulation to make it handle both fully and weakly annotated data simultaneously. Our multi-utility SSVM is formally defined as follows:

$$\min_{\mathbf{w},\boldsymbol{\xi}\geq\mathbf{0},\boldsymbol{\eta}\geq\mathbf{0}} \frac{1}{2}\mathbf{w}^\intercal\mathbf{w} + \frac{C}{N+M}\left(\sum_{n=1}^{N}\xi_n + \alpha\sum_{m=1}^{M}\eta_m\right), \tag{5}$$

$$\text{s.t.} \qquad F(\mathbf{x}_n, \mathbf{y}_n; \mathbf{w}) \geq \max_{\bar{\mathbf{y}}\in\mathcal{Y}_n}\left(F(\mathbf{x}_n, \bar{\mathbf{y}}; \mathbf{w}) + \Delta(\bar{\mathbf{y}}, \mathbf{y}_n)\right) - \xi_n, \quad \forall n, \tag{6}$$

$$\max_{\mathbf{y}\in\mathcal{L}(\mathbf{z}_m)} F(\mathbf{x}_m, \mathbf{y}; \mathbf{w}) \geq \max_{\bar{\mathbf{y}}\in\mathcal{Y}_m}\left(F(\mathbf{x}_m, \bar{\mathbf{y}}; \mathbf{w}) + \mathrm{K}(\bar{\mathbf{y}}, \mathbf{z}_m)\right) - \eta_m, \forall m. \tag{7}$$

Note that for $M = 0$ the above formulation degenerates to the standard SSVM formulation, while for $N = 0$ it reduces to the latent-variable SSVM [24]. Note also that the full labelling $\mathbf{y}_n$ can be seen as a degenerate weak annotation, where $\mathcal{L}(\mathbf{z}_m) = \{\mathbf{y}_n\}$. Therefore, Problem (5)–(7) is almost equivalent to LV-SSVM, but it contains the slack balancing coefficient α. Ignoring this coefficient may hurt the performance of multi-utility learning, as we show in Section 4.2. In order to perform the optimization, in addition to the loss-augmented inference in (6), we should also be able to perform the weak-loss augmented inference in (7), as well as the *annotation-consistent inference* in the left-hand side of (7).

Optimization problem (5)–(7) is not convex and thus hard. We follow Yu and Joachims [24] and use the concave-convex procedure (CCCP) [25] to solve it approximately.

3 Weak annotation for semantic image segmentation

Semantic image segmentation aims to assign category labels to image pixels. We assume that an image is represented as a set of *superpixels*, i.e. groups of co-located pixels similar by appearance. Consider a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. Its nodes $\mathcal{V}$ correspond to superpixels of the image. The set of edges $\mathcal{E}$ represents a neighborhood system on $\mathcal{V}$ that includes the pairs of nodes that correspond to all adjacent superpixels. Let $\mathbf{x}_i \in \mathbb{R}^d$ be a vector of superpixel features associated with some node $i \in \mathcal{V}$, $\mathbf{x}_{ij} \in \mathbb{R}^e$ be a vector of superpixel interaction features for the edge connecting nodes i and j, and $\mathbf{x} = \bigoplus_{i\in\mathcal{V}}\mathbf{x}_i \oplus \bigoplus_{(i,j)\in\mathcal{E}}\mathbf{x}_{ij}$ be their concatenation. The value of each variable y_i corresponds to the label of the i-th superpixel. We use the following discriminant function F:

$$F(\mathbf{x}, \mathbf{y}; \mathbf{w}) = \mathbf{w}^\intercal\boldsymbol{\Psi}(\mathbf{x}, \mathbf{y}) = \sum_{i\in\mathcal{V}}\sum_{k=1}^{K}[y_i = k](\mathbf{x}_i^\intercal\mathbf{w}_k^{\mathrm{u}}) + \sum_{(i,j)\in\mathcal{E}}[y_i = y_j](\mathbf{x}_{ij}^\intercal\mathbf{w}^{\mathrm{p}}), \tag{8}$$

where $\mathbf{w} = \bigoplus_{k=1}^{K}\mathbf{w}_k^{\mathrm{u}} \oplus \mathbf{w}^{\mathrm{p}}$ is a vector of the model parameters, and $\mathbf{w}_k^{\mathrm{u}} \in \mathbb{R}^d$, $\mathbf{w}^{\mathrm{p}} \in \mathbb{R}^e$. The summands in the first and the second terms are called unary and pairwise potentials, respectively. We restrict pairwise weights $\mathbf{w}^{\mathrm{p}}$ and pairwise features $\mathbf{x}_{ij}$ to be nonnegative and thus obtain an associative discriminative function (with only attractive pairwise potentials) [17]. Maximizing $F(\mathbf{x}, \mathbf{y}; \mathbf{w})$ w.r.t. $\mathbf{y}$ is known to be NP-hard, but efficient approximate algorithms exist, e.g. α-expansion [2].

We use the weighted Hamming loss (4) for fully-labelled images, where c_i is the number of pixels in the corresponding superpixel, so the loss function estimates the number of mislabelled image pixels.[2] To use some type of weak annotations for training, we need to define the annotation-specific loss function that allows loss-augmented inference and annotation-consistent inference. The

[2] In practice, ground-truth labelling of a superpixel may contain several labels; in this case the number of incorrectly inferred pixels is added to the loss. We ignore this case to ease the notation, but all the algorithms still work in that case.

former should be efficient, since it is performed in the inner loop of training and thus is typically a bottleneck. We show how to define and combine them for the annotations of the following types: image-level labels, bounding boxes around objects, and objects' seeds.

3.1 Image-level labels

We start by defining loss functions $\mathrm{K}(\mathbf{y}, \mathbf{z})$ for some arbitrary labelling $\mathbf{y}$ and ground-truth weak annotation $\mathbf{z}$. In this subsection we assume that $\mathbf{z}$ is a set of labels used in the ground-truth image labelling (for the image in Fig. 1, $\mathbf{z} = \{$'sky', 'tree', 'plain', 'grass'$\}$). We cannot compute the Hamming loss (4) if the full labelling is unknown for one of its arguments. Let's instead define a proxy loss function, that is symmetric and does not compare labels of any superpixels directly:

$$\Delta_{\text{il}}(\mathbf{y}, \bar{\mathbf{y}}) = \sum_{i \in \mathcal{V}} c_i [\nexists j \in \mathcal{V} : y_j = \bar{y}_i \ \vee \ \nexists j \in \mathcal{V} : \bar{y}_j = y_i]. \tag{9}$$

It penalizes all the superpixels that have been given any label that lacks in the annotation $\bar{\mathbf{y}}$, as well as superpixels which have ground truth labels that lack in $\mathbf{y}$. Unfortunately, the ground-truth labelling $\bar{\mathbf{y}}$ is unknown. If we knew the areas S_k of each label $k \in \mathbf{z}$, we could derive the following upper bound on (9):

$$\mathrm{K}_{\text{il}}(\mathbf{y}, \mathbf{z}; \{S_k\}_{k \in \mathbf{z}}) = \sum_{k \notin \mathbf{z}} \sum_{i \in \mathcal{V}} c_i [y_i = k] + \sum_{k \in \mathbf{z}} S_k \prod_{i \in \mathcal{V}} [y_i \neq k]. \tag{10}$$

This upper bound is tight up to a factor of 2. The first term penalizes the pixels labelled with wrong labels, while the second term penalizes ignoring the labels from $\mathbf{z}$.

Since we do not know the areas S_k, the best we can do is to assume $\mathrm{K}(\mathbf{y}, \mathbf{z})$ to be the expectation of (10) taken over all full labellings consistent with $\mathbf{z}$. If there are enough fully-labelled images, the areas S_k can be estimated. Otherwise we assume the uniform distribution over the feasible full labellings $\mathbf{y} \in \mathbf{z}$ and get

$$\mathrm{K}_{\text{il}}(\mathbf{y}, \mathbf{z}) = \sum_{k \notin \mathbf{z}} \sum_{i \in \mathcal{V}} c_i [y_i = k] + \sum_{k \in \mathbf{z}} \frac{\sum_{i \in \mathcal{V}} c_i}{|\mathbf{z}|} \prod_{i \in \mathcal{V}} [y_i \neq k]. \tag{11}$$

Having defined the loss function K_{il}, we need to provide algorithms for inference problems in (7). For annotation-consistent inference $\max_{\mathbf{y} \in \mathbf{z}_m} F(\mathbf{x}_m, \mathbf{y}; \mathbf{w})$ we use α-expansion over the labels from $\mathbf{z}_m$ only. Note that this may result in an inconsistent labelling: some labels from $\mathbf{z}_m$ may miss in $\mathbf{y}$. We have tried an heuristic algorithm for making it strictly consistent with $\mathbf{z}$ by changing one node per missing label, but observed no significant difference in practice.

The loss-augmented inference is now not decomposable to unary and pairwise factors. To work this around, we derive:

$$\max_{\bar{\mathbf{y}} \in \mathcal{Y}_m} \left(F(\mathbf{x}_m, \bar{\mathbf{y}}; \mathbf{w}) + K_{\text{il}}(\bar{\mathbf{y}}, \mathbf{z}_m)\right) =$$

$$\max_{\bar{\mathbf{y}} \in \mathcal{Y}_m} \left(F(\mathbf{x}_m, \bar{\mathbf{y}}; \mathbf{w}) + \sum_{k \notin \mathbf{z}} \sum_{i \in \mathcal{V}} c_i [\bar{y}_i = k] - \sum_{k \in \mathbf{z}} \frac{\sum_{i \in \mathcal{V}} c_i}{|\mathbf{z}|} [\exists i : \bar{y}_i = k] \right) + \text{const}. \tag{12}$$

The last maximization is the standard MRF inference problem with label costs. We use the efficient modification of α-expansion for accounting label costs [4].

3.2 Bounding boxes

It is convenient to annotate instances in an image with tight bounding boxes (Fig. 1c). On the other hand, they do not give much information for background categories. Therefore, we consider the annotation that consists of both bounding boxes and image-level labels. For example, annotation of an image may contain the bounding-box locations of cars and pedestrians, and additionally state that there are buildings, road, and sky in the image. We assume that within a certain image each category can be defined either with image-level labels, or with bounding boxes, though the type of annotation for a category may vary from image to image (see Section 4.3 for an example where it can be useful).

We model weak annotation $\mathbf{z}$ of an image as a pair $(\mathbf{z}^{\text{il}}, \mathbf{z}^{\text{bb}})$ of image-level and bounding box annotations. The latter is a set of bounding boxes with associated category labels: $\mathbf{z}^{\text{bb}} = \{z_i\}$, with the following functions defined: $\mathit{label}(z_i)$, which defines the associated category label, and $\mathit{box}(z_i) = [\mathit{left}(z_i), \mathit{right}(z_i)] \times [\mathit{top}(z_i), \mathit{bottom}(z_i)]$ that defines the extent of the bounding box. The set of labels $\mathcal{K}$ is partitioned into three subsets w.r.t. the weak annotation $\mathbf{z}$: the labels that are defined with bounding boxes ($\mathcal{K}_b = \bigcup_{z \in \mathbf{z}^{\text{bb}}} \mathit{label}(z)$), those that are present somewhere else in the image ($\mathcal{K}_p = \mathbf{z}^{\text{il}}$), and those that are absent ($\mathcal{K}_a = \mathcal{K} \setminus (\mathcal{K}_b \cup \mathcal{K}_p)$). Nodes $\mathcal{V}$ are also partitioned: $\mathcal{V}_k = \bigcup_{z \in \mathbf{z}^{\text{bb}}: \mathit{label}(z)=k} \mathit{box(z)}$ is the union of pixel indices in the bounding boxes corresponding to the label $k \in \mathcal{K}_b$, and $\mathcal{V}_0 = \mathcal{V} \setminus \bigcup_{k \in \mathcal{K}_b} \mathcal{V}_k$. We now define the combined loss function as:

$$\mathrm{K}_{\text{il-bb}}(\mathbf{y}, \mathbf{z}) = \sum_{k \in \mathcal{K}_a} \sum_{i \in \mathcal{V}} c_i \left[y_i = k\right] + \sum_{k \in \mathcal{K}_p} \sigma_k \prod_{i \in \mathcal{V}} \left[y_i \neq k\right] + \beta \sum_{z \in \mathbf{z}^{\text{bb}}} \left(\sum_{p=\mathit{top}(z)}^{\mathit{bottom}(z)} \nu_p^z \prod_{q=\mathit{left}(z)}^{\mathit{right}(z)} V((p,q); \mathbf{y}, \mathit{label}(z)) + \sum_{q=\mathit{left}(z)}^{\mathit{right}(z)} \omega_q^z \prod_{p=\mathit{top}(z)}^{\mathit{bottom}(z)} V((p,q); \mathbf{y}, \mathit{label}(z)) \right) + \sum_{k \in \mathcal{K}_b} \sum_{i \in \mathcal{V}_0} c_i \left[y_i = k\right]. \quad (13)$$

The first two terms are almost the same as in (11), but the estimate of the category area in the second term does not include the pixels within the bounding boxes: $\sigma_k = \left(\sum_{i \in \mathcal{V}_0} c_i\right) / |\mathbf{z}^{\text{il}}|$. The third term penalizes 'empty' rows and columns in the bounding boxes, i.e. those rows and columns that do not contain pixels of a target category at all. The violation function V is defined as:

$$V(\mathbf{p}; \mathbf{y}, k) = \begin{cases} 1, & \text{if } \mathit{map}(\mathbf{y})_{\mathbf{p}} \neq k, \\ 0, & \text{otherwise.} \end{cases} \quad (14)$$

Here $\mathit{map}(\mathbf{y})$ is the classification map induced by the superpixel labelling $\mathbf{y}$. Coefficients ν_p^z and ω_q^z allow us to assign the penalty for the corresponding row or column being empty, depending on its position in the bounding box. One can learn the category-specific profiles of ν^z and ω^z when the full labelling is abundant enough, but we use uniform profiles assuming that half of a bounding box is occupied by the object on average: $\nu_p^z = \big(\mathit{right}(z) - \mathit{left}(z)\big)/2$, $\omega_q^z = \big(\mathit{bottom}(z) - \mathit{top}(z)\big)/2$. Note that this makes the loss an estimate on the number of mislabelled pixels (similar to the image-level label loss (11)), so the value coefficient $\beta = 1$ should work well (we show in Section 4.3 that it really does). We have also tried linearly decreasing loss used by Kumar et al. [8], but it did not affect the performance significantly. The last term penalizes the bounding-box labels outside of bounding boxes.

We have shown in the previous section how to account for the two initial terms in the loss-augmented inference. The last term is decomposable w.r.t. superpixels. The third term is a sum over the higher-order cliques of the following form. For each bounding box z, each row and each column generates a clique of nodes corresponding to the superpixels that intersect that row/column. We treat them the same way as the image-level loss: we modify α-expansion with label costs [4] to penalize each clique of superpixels, which contains at least one superpixel labelled with $\mathit{label}(z)$. There is a technical difficulty with the superpixels that cross the bounding box border: it is unclear if their labelling with $\mathit{label}(z)$ should be penalized. We adopted the following strategy: shrink the bounding box to allow some margin, and treat all superpixels that intersect the shrunk bounding box (and only them) as insiders. We set the margin width equal to 6% of the corresponding bounding box dimension.

During the annotation-consistent inference, we need to infer a labelling that has objects only in bounding boxes of the corresponding category labels, and they should fill those bounding boxes tightly, i.e. touch upon all four sides of the bounding box shrunk to allow a 6% margin (Lempitsky et al. [9] showed that this definition is natural). The first condition is easy to satisfy: we can suppress certain labels outside of bounding boxes by using infinite unary potentials. To provide tightness, we use a variation of the pinpointing algorithm [9], adapted for the multi-class segmentation. First, segmentation is performed without the tightness constraints. Then, until those constraints are satisfied, one of the superpixels changes its unary potential, and expansion move is performed. In our implementation, we select the superpixel with the highest relative potential for $\mathit{label}(z)$ that has not been assigned this label yet, and assign it the infinite potential for $\mathit{label}(z)$ to guarantee that it will change

its label. This procedure is finite because at each iteration at least one superpixel within $box(z)$ switches to $label(z)$. In contrast to Lempitsky et al. [9], we do not perform further dilation, since it is unclear, which label we should use for expansion move(s); neither of the heuristics we tried improved the result significantly. We also found that initialization of the latent variables in LV-SSVM matters: we obtained the best results when initially all superpixels within $box(z)$ were initialized with $label(z)$. Note that Kumar et al. [8] used a different criterion during the annotation-consistent inference: they penalize the empty rows and columns within bounding boxes (the opposite to what we do in loss-augmented inference). Note that their heuristic does not guarantee the tightness of the resulting segmentation.

3.3 Objects' seeds

Another form of a weak annotation natural for the object categories is the seed annotation (Fig. 1d). In general, for a segment of some category, a seed is a subset of its pixels. We consider a particular case, where only one pixel, persumably close to the segment center, is labelled. During the annotation-consistent inference, we require the superpixel where this point is located to have the fixed seed label.

We now model the weak annotation $\mathbf{z}$ as a pair $(\mathbf{z}^{\text{il}}, \mathbf{z}^{\text{os}})$, where $\mathbf{z}^{\text{os}}$ is a set of 2D points with the corresponding labels: $(\mathbf{p}, k)$. The seed centrality assumption allows us to set the Gaussian penalty for inferring any non-seed label in the neighbourhood of each seed, which brings us to the following loss function:

$$\mathrm{K}_{\text{il-os}}(\mathbf{y}, \mathbf{z}) = \sum_{k \in \mathbf{k}_a} \sum_{i \in \mathcal{V}} c_i \left[y_i = k\right] + \sum_{k \in \mathbf{k}_p} \sigma_k \prod_{i \in \mathcal{V}} \left[y_i \neq k\right] + \beta \sum_{\substack{(\mathbf{p}', k') \\ \in \mathbf{z}^{\text{os}}}} \sum_{\mathbf{p} \in I} V(\mathbf{p}; \mathbf{y}, k') \exp\left(-\frac{\pi \|\mathbf{p} - \mathbf{p}'\|^2}{\tau_{k'}}\right). \quad (15)$$

Here the first two terms are the same as in the image-level label loss. The inner sum in the third term is taken over all image pixels I. The form of the Gaussian is defined in such a way that the penalty for misclassification of the central pixel $\mathbf{p}'$ is 1, and whenever no superpixels of the label k' are found, the penalty is equal to the estimated area of the label k' w.r.t. all labellings consistent with the weak annotation; specifically,

$$\tau_{k'} = \frac{\sum_{i \in \mathcal{V}} c_i}{(|\mathbf{z}^{\text{il}}| + \#Lab(\mathbf{z}^{\text{os}})) \cdot \#Obj(\mathbf{z}^{\text{os}}, k')}. \quad (16)$$

Here $\#Lab(\mathbf{z}^{\text{os}})$ is the number of different labels in $\mathbf{z}^{\text{os}}$, and $\#Obj(\mathbf{z}^{\text{os}}, k')$ is the number of seeds of the label k' in $\mathbf{z}^{\text{os}}$. Loss (15) is decomposable to factors, so the loss-augmented inference is trivial.

4 Experiments

4.1 Datasets and metrics

We test the proposed framework on two datasets: MSRCv2[3] [15, 22] and SIFT-flow[4] [10, 18, 23]. MSRC contains 276 training and 256 test images that are fully labelled using 23 category labels; significant part of pixels remains unlabelled. SIFT-flow is a more challenging dataset: it is a subset of the LabelMe database [19], which contains 2488 training and 200 test images; they are labelled to 33 categories using crowd-sourcing.

For *MSRC*, we obtain superpixels using the original implementation of the *gPb* edge detector [1]. The unary features are the following: a histogram of SIFT [12] visual words built using a dictionary of size 512 by hard assignment of the descriptors to the bins; a histogram of the RGB colors on a dictionary of size 128; a histogram of locations over a uniform 6×6 grid. We L_2-normalize the joint feature vector and map it into a higher-dimensional space where the inner product approximates the χ^2-kernel in the original space (the dimensionality of the space triples after the transformation) [21].

[3] http://research.microsoft.com/en-us/projects/objectclassrecognition/
[4] http://people.csail.mit.edu/celiu/LabelTransfer/code.html

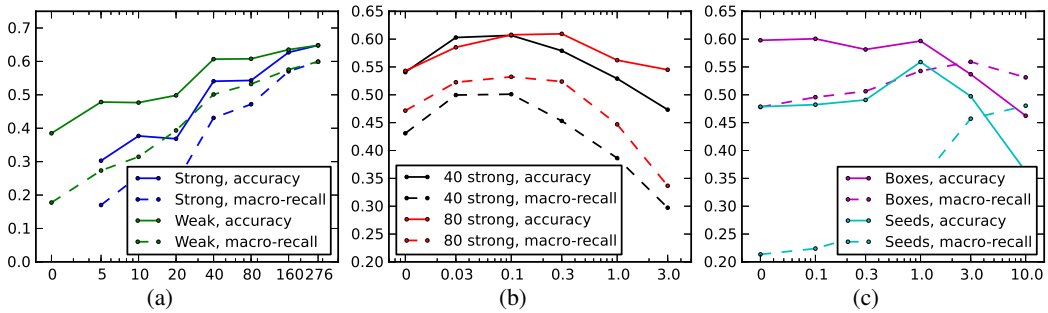

Figure 2: (a)–(c) Accuracy (solid lines) and per-class recall (dashed lines) subject to different parameters on the MSRC dataset. (a) Varying the number of fully-labelled images. Blue line show test set segmentation quality when only fully-labelled images are available; green line—when the complementary part of the train set has image-level labels. (b) Varying the coefficient of the weak-loss coefficient α. Black line show test set segmentation quality when 40 images are fully labelled, red line—when 80 images; the complementary part of the train set has image-level labels. (c) Varying the coefficient of the bounding box (magenta line) or object seed (cyan line) loss β. All 276 training images have image-level labels, all objects have tight bounding box or seed annotations, respectively

We use pairwise factors over the pairs of the superpixels that share a common border and use the following 4 pairwise features: $\exp(-c_{ij}/10)$, $\exp(-c_{ij}/40)$, $\exp(-c_{ij}/100)$, 1. Here c_{ij} is the strength of the boundary between segments i and j returned by the *gPb*.

For *SIFT-flow*, we follow Vezhnevets et al. [23] and obtain superpixels and features using the code by Tighe and Lazebnik [18]. It runs graph-based segmentation of Felzenszwalb and Huttenlocher [5] followed by feature extraction. The unary features include shape, location, texture, color and appearance feature vectors, some of which are also computed over dilated superpixel masks to capture the context: 3115 unary features in total. We also transform this vector with a χ^2-kernel approximation, which triples its size. We use pairwise factors over the pairs of superpixels that share a common border and the pairwise features computed as distances between groups of superpixels' features (χ^2 distance in case of histograms, Euclidean otherwise), 26 features in total.

Quality measures. We use two standard measures of segmentation quality: accuracy and per-class recall. The accuracy is defined as the rate of correctly labelled pixels of the test set. The per-class recall is the number of correctly labelled pixels of each category divided by the true total area of that category, averaged over categories. Following the previous work [22, 14], we exclude the pixels of rare categories ('horse' and 'mountain') from recall computation for MSRC, but include the 'other' label, see Section 4.2. Similarly, we exclude rare categories ('cow', 'desert', 'moon', and 'sun') from SIFT-flow recall computation.

4.2 Image-level labels

Generating weak annotation. We obtain image-level labels automatically from full labellings by enumerating the unique labels for each image. Each MSRC image typically features one or several objects of some target category (e.g. 'sign', 'cow', 'car') on top of some background. Not every background category falls into the used labels, so it may remain unlabelled. Thus, some images contain only one category label. In this case the image-level label unambiguously defines the full labelling. To avoid this knowledge (unrealistic in the real-world setting), we could model the 'other' label, which contains anything but the labelled 23 categories. However, the labellings typically have uncertain borders between segments of different labels, i.e. the borders are unlabelled too (Fig. 1b). If we modelled those boundaries as a separate category, it would hurt the segmentation performance. Instead, we want to model this 'other' label only for unlabelled regions, not for the boundaries. We use the following heuristic criterion for obtaining image-level labels: if an image contains only one label, or at least 30% of its pixels are unlabelled, we include them to the image-level label as the 'other' label.

Varying the full-labelling rate. In our basic setting we have a (possibly empty) part of the training set fully labelled, while the rest of the images have only image-level labels. We generate those

Table 1: Accuracy and average per-class recall on the SIFT-flow dataset. The first two lines describe training on the subset of 256 fully labelled images of the models with and without pairwise potentials, respectively. The third line experiment used the whole dataset with image-level labels, but for only 256 of them full labelling is known. The bottom line shows the result when the whole dataset is fully labelled

experiment	acc	rec
256/256 strong, local	0.574	0.167
256/256 strong, init loc.	0.620	0.176
256/2488 strong, init ↑	0.674	0.208
2488/2488 strong	0.696	0.246

Table 2: Accuracy (first number in each cell) and average per-class recall (second number) on the MSRC dataset when during training i) only full labelling is available, ii) image-level (il) labels are also available for the rest of the data set, iii) object seeds (os) are additionally available, iv) bounding boxes (bb) for objects are available, v) both seeds and bounding boxes are available. Note that the numbers in the last column are all equal since the weak annotation does not add any information when all training set is fully labelled

il	bb	os	0/276 strong	5/276 strong	276 strong
−	−	−	n/a	0.300/0.170	0.648/0.599
+	−	−	0.385/0.178	0.478/0.273	0.648/0.599
+	−	+	0.559/0.346	0.574/0.370	0.648/0.599
+	+	−	0.597/0.543	0.606/0.546	0.648/0.599
+	+	+	0.531/0.567	0.542/0.564	0.648/0.599

subsets using the Metropolis–Hastings sampling, trying to make the distribution of their label counts approximate that of the whole training set. Fig. 2a shows the accuracy and per-class recall of the test set segmentation for various full labelling rates in comparison to the fully-supervised setting.[5] In the most common scenario—when less than 20% of the training set is fully labelled—the weakly-annotated subset provides a stable 10–15% improvement both in terms of the accuracy and mean per-class recall.

Balancing the loss functions. When the training set consists of both weak annotations and full labellings, the coefficient α from (5) needs to be set. We discovered that its optimal value was lower than 1 (Fig. 2b shows the dependency of performance on α). We speculate that this is because we are more certain about the strong loss, so it should contribute to the objective more. Thus, for all the other experiments we set $\alpha = 0.1$.

SIFT-flow results. On the SIFT-flow dataset, we compare fully-supervised learning with weakly-supervised at one point, i.e. when only 256 training images are fully labelled, and the rest 2232 images have only image-level labels (Table 1). This weakly-learned model loses to the fully-supervised one only 2% in the accuracy and 4% in the per-class recall. Note that our model is *on par* with Vezhnevets et al. [23], who also reached 21% on that dataset with the same superpixels and features. The difference is they used only image-level annotation, while we used about 10% fully labelled images. However, their model is substantially more complicated: they use extremely-randomized hashing forest for non-linear feature transform, learn objectness and image-level priors, and connect superpixels of different images within the multi-image model. Since the LV-SSVM optimization problem is not convex, the algorithm may get stuck at local minima. We initialize the parameters of LV-SSVM by the parameters of the SSVM trained on the fully-labelled part of the dataset, if there is one.

4.3 Adding bounding boxes and seeds

Generating weak annotation. We generate two more annotations for the MSRC training data to test additional annotation-specific loss functions. Similar to image-level labels, we generate them from the full labelling. Tight bounding boxes and object seeds are good for description of the object ('thing') categories, while do not add much information beyond image-level labels for the background ('stuff') categories. We divide the list of categories into two parts: background, which includes 'grass', 'sky', 'mountain', 'water', 'road', and 'other'; and objects, which includes all other categories. There are two ambivalent categories—'building' and 'tree'—which can instantiate either a target object of a photograph, or background. We used the following heuristic for each image: consider tree and building as background iff there are other objects in the image. We enhanced the image-level labelling with either tight bounding boxes or object seeds for segments of object categories only. For the other categories, only image-level labels were available. To generate seeds,

[5] http://shapovalov.ro/data/MSRC-weak-train-masks.zip

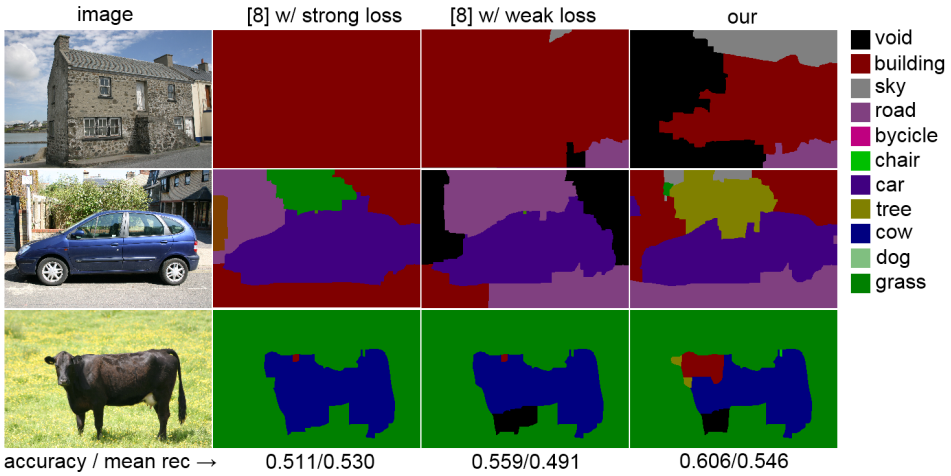

Figure 3: Qualitative results of the proposed algorithm and two variations of the algorithm by Kumar et al. [8] applied to three images from the MSRC test set

for each segment we took its pole of inaccessibility—the point that maximizes its distance transform map.

Results. Table 2 summarizes the results. When the full labelling is unavailable, both object seed and bounding box annotations give significant improvement over just image-level labels. Bounding boxes notably increase per-class recall: they help to better learn 'thing' categories, which are numerous and typically have smaller area. Overall, learning with bounding boxes only 5% inferior to learning on fully labelled data both in terms of the accuracy and per-class recall. Object seed annotation gave more modest increase in performance, though is easier to obtain. We used the value $\beta = 1$ to balance the impact of image-level vs. bounding box (or seed) loss functions: they seem to provide equal contribution to the objective function; Fig. 2c supports that hypothesis.

Comparison to Kumar et al. [8]. Unfortunately, we cannot directly compare to Kumar et al. [8] since the type of input data for their framework is unorthodox. They use two different datasets to obtain segmentation maps (partial labellings) for the foreground and background categories, respectively. Our framework does not support this kind of annotation: we believe that it is easier to obtain segmentation for background and foreground categories using the same set of images. This eliminates the need to use the latent-variable SSVM for training the basic model; instead the global minimum of SSVM objective can be found efficiently. Also, when both image-level labels and bounding boxes (or seeds) are known for each weakly-annotated image, both background and foreground partial labellings can be inferred, and using latent-variable SSVM after adding weakly-annotated data is not necessary again. Thus, when given the data we use, the method of Kumar et al. [8] could look like this:

- train SSVM using the fully-labelled part of the training set,
- use the trained model to infer the labelling of all images consistent with the weak annotation,
- train SSVM using the hallucinated labelling obtained in the previous step.

This method is similar to running one outer iteration of our training algorithm, but it has one important difference: the loss function in the second SSVM. While our method uses the weak loss function, the modified method of Kumar et al. [8] uses the strong loss function w.r.t. the hallucinated labelling. To compare the methods, we use the MSRC training set with 5 fully-labelled images and the rest annotated with bounding boxes and image-level labels (row 4, column 2 in Table 2, excluding headers) to train both described modifications: with the weak bounding-box loss function (13), and with the strong loss function (4) (still different from the loss function of Kumar et al. [8]). The segmentation maps and numerical results in Fig. 3 show that the proposed simultaneous minimization of loss functions is superior both in terms of accuracy and per-class recall.

5 Conclusion

We presented the framework for learning structural classification from different types of annotations by minimizing annotation-specific loss functions. We applied it to semantic image segmentation by introducing weak loss functions for for image-level, bounding box, and object seed annotations. Usage of weakly-annotated training data consistently improves the labelling. The results on the semantic segmentation datasets show that the joint annotation where background is given by image-level labels, and objects are given by bounding boxes, is the best trade-off between segmentation quality and annotation effort.

References

[1] Arbeláez, P., Maire, M., Fowlkes, C., and Malik, J. (2011). Contour detection and hierarchical image segmentation. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 33(5):898–916.

[2] Boykov, Y., Veksler, O., and Zabih, R. (2001). Fast approximate energy minimization via graph cuts. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 23(11):1222–1239.

[3] Chang, M.-W., Srikumar, V., Goldwasser, D., and Roth, D. (2010). Structured output learning with indirect supervision. In *International Conference on Machine Learning*.

[4] Delong, A., Osokin, A., Isack, H. N., and Boykov, Y. (2012). Fast Approximate Energy Minimization with Label Costs. *International Journal of Computer Vision*, 96(1):1–27.

[5] Felzenszwalb, P. F. and Huttenlocher, D. P. (2004). Efficient Graph-Based Image Segmentation. *International Journal of Computer Vision*, 59(2):167–181.

[6] Heitz, G. and Koller, D. (2008). Learning spatial context: Using stuff to find things. In *European Conference on Computer Vision*, pages 30–43, Marseille, France. Springer.

[7] Joachims, T., Finley, T., and Yu, C. (2009). Cutting-plane training of structural SVMs. *Machine Learning*, 77(1):27–59.

[8] Kumar, M. P., Turki, H., Preston, D., and Koller, D. (2011). Learning specific-class segmentation from diverse data. In *IEEE International Conference on Computer Vision*, pages 1800–1807.

[9] Lempitsky, V., Kohli, P., Rother, C., and Sharp, T. (2009). Image segmentation with a bounding box prior. In *International Conference on Computer Vision*, pages 277–284.

[10] Liu, K., Raghavan, S., Nelesen, S., Linder, C. R., and Warnow, T. (2009). Rapid and accurate large-scale coestimation of sequence alignments and phylogenetic trees. *Science (New York, N.Y.)*, 324(5934):1561–4.

[11] Lou, X. and Hamprecht, F. A. (2012). Structured Learning from Partial Annotations. In *International Conference on Machine Learning*.

[12] Lowe, D. G. (2004). Distinctive Image Features from Scale-Invariant Keypoints. *International Journal of Computer Vision*, 60(2):91–110.

[13] Pletscher, P. and Kohli, P. (2012). Learning low-order models for enforcing high-order statistics. In *International Conference on Artificial Intelligence and Statistics*.

[14] Shotton, J., Johnson, M., and Cipolla, R. (2008). Semantic texton forests for image categorization and segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*.

[15] Shotton, J., Winn, J., Rother, C., and Criminisi, A. (2006). Textonboost: Joint appearance, shape and context modeling for multi-class object recognition and segmentation. In *European Conference on Computer Vision*, pages 1–14.

[16] Tarlow, D. and Zemel, R. S. (2012). Structured Output Learning with High Order Loss Functions. In *International Conference on Artificial Intelligence and Statistics*.

[17] Taskar, B., Chatalbashev, V., and Koller, D. (2004). Learning associative Markov networks. In *International Conference on Machine Learning*, pages 102–109, Banff, Alberta, Canada.

[18] Tighe, J. and Lazebnik, S. (2010). SuperParsing: Scalable Nonparametric Image Parsing with Superpixels. In *European Conference on Computer Vision*, Heraklion, Grece.

[19] Torralba, A., Russel, B. C., and Yuen, J. (2010). LabelMe: Online Image Annotation and Applications. *Proceedings of the IEEE*, 98(8):1467–1484.

[20] Tsochantaridis, I., Joachims, T., Hofmann, T., and Altun, Y. (2006). Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484.

[21] Vedaldi, A. and Zisserman, A. (2010). Efficient Additive Kernels via Explicit Feature Maps. In *IEEE Conference on Computer Vision and Pattern Recognition*, San-Francisco, CA.

[22] Vezhnevets, A., Ferrari, V., and Buhmann, J. M. (2011). Weakly Supervised Semantic Segmentation with a Multi-Image Model. In *IEEE International Conference on Computer Vision*, Barcelona, ES.

[23] Vezhnevets, A., Ferrari, V., and Buhmann, J. M. (2012). Weakly Supervised Structured Output Learning for Semantic Segmentation. In *IEEE Conference on Computer Vision and Pattern Recognition*, Providence, RI.

[24] Yu, C.-N. J. and Joachims, T. (2009). Learning structural SVMs with latent variables. In *International Conference on Machine Learning*, Montreal, Canada.

[25] Yuille, A. and Rangarajan, A. (2002). The concave-convex procedure (CCCP). In *NIPS*.