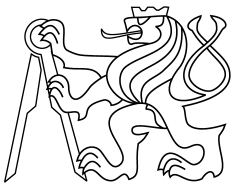
CENTER FOR
MACHINE PERCEPTION



CZECH TECHNICAL
UNIVERSITY IN PRAGUE

RESEARCH REPORT

ISSN 1213-2365

Discriminative structured output learning from partially annotated examples

(Version 1.0)

Vojtěch Franc, Konstiantyn Antoniuk, Michal Uřičář

{xfrancv,antonkos,uricamic}@cmp.felk.cvut.cz

CTU–CMP–2012–16

July 1, 2012

This research was supported by a Marie Curie European Reintegration Grant SEMISOL (PERG04-GA-2008-239455) within the 7th European Community Framework Programme.

Research Reports of CMP, Czech Technical University in Prague, No. 16, 2012

Published by

Center for Machine Perception, Department of Cybernetics
Faculty of Electrical Engineering, Czech Technical University
Technická 2, 166 27 Prague 6, Czech Republic
fax +420 2 2435 7385, phone +420 2 2435 7637, www: http://cmp.felk.cvut.cz

Discriminative structured output learning from partially annotated examples

Vojtěch Franc, Konstiantyn Antoniuk, Michal Uřičář

July 1, 2012

Abstract

The discriminative structured output learning has been proved successful in solving many real-life applications. A big deficiency of existing algorithms like the Structured Output SVMs is the requirement of fully annotated training examples. In this report we formulate a problem of learning the structured output classifiers from partially annotated examples as an instance of the expected risk minimization. We show that the minimization of the expected risk is equivalent to the minimization of the partial loss which can be evaluated on partially annotated examples. We proposed an instance of the partial learning algorithm for the class of linear structured output classifiers which we call Partial-SO-SVM. The Partial-SO-SVM algorithm leads to a hard non-convex optimization problem. We provide an algorithm solving the Partial-SO-SVM problem approximately using an additional prior knowledge about the problem. We demonstrated effectiveness of the proposed method on two real life computer vision problems, namely, the face landmark detection and the image segmentation.

1 Introduction

The discriminative structured output learning has been proved successful in solving many real-life applications. Among the most successful is the Structured Output Support Vector Machines (SO-SVM) algorithm [4] which translate learning of the linear structured output classifier into a convex optimization tasks. A big deficiency of the SO-SVM is the requirement of fully annotated training examples. Annotation of data for the structured output learning is often tedious and expensive. There is a strong demand for an extension of the existing supervised SO-SVMs for learning from the partially annotated examples.

Recently, there has been several attempts in this direction. The paper of [3] proposes a convex formulation of learning from ambiguously annotated examples for the case of flat (unstructured) classifiers. Unfortunately, it is not clear how to extend their approach to the structured output setting. A large margin formulation of the structured output learning from partially annotated examples has been proposed in [6]. Their approach is based on the minimization of a partial loss which takes into account only the labels provided in the partial annotation. The approach has two deficiencies. First, the minimization of the partial loss has no theoretical justification. Second, unlike the fully supervised SO-SVM the learning problem is not convex and, currently, only naive optimization methods finding only a local optima exist.

The main contribution of this report is in proving a clear statistical formulation of the problem of learning the structured output classifiers from partially annotated examples. We formulate learning as an instance of the expected risk minimization. We show that the minimization of the expected risk is equivalent to the minimization of the partial loss which can be evaluated on partially annotated examples. We propose an instance of the partial learning for the class of linear structured output classifiers which we call Partial-SO-SVM. The Partial-SO-SVM algorithm leads to a hard non-convex optimization problem similar to that of [6] though their partial loss is slightly different. As a second contribution, we provide an algorithm solving the Partial-SO-SVM problem approximately using an additional prior knowledge about the problem. The approximate algorithm transforms the original partial learning problem to a series of simpler problems resembling the standard supervised SO-SVM which allows usage of existing solvers.We demonstrate effectiveness of the proposed method on two real life computer vision problems, namely, the face landmark detection and the image segmentation.

2 Discriminative fully supervised learning

In this section we formulate a known problem of discriminative learning of the linear structured output classifiers from fully annotated examples. We denote this setting as the *fully supervised learning*. The purpose of this section is to introduce notation used in this report in order to synchronize with existing literature.

Let $\boldsymbol{y} = (y_t \in \mathcal{Y}_t \mid t \in \mathcal{T}) \in \mathcal{Y}$ be a labeling assigned to a set of objects $t \in \mathcal{T}$. Let $\boldsymbol{x} \in \mathcal{X}$ be a vector of observations [1] and $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \Re$ a given

[1] For simplicity we assume that the set $\mathcal{X}$ is finite but the extension to the continuous case is straightforward.

loss function.

Given a training multi-set $\{(\boldsymbol{x}^1, \boldsymbol{y}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{y}^m)\} \in (\mathcal{X} \times \mathcal{Y})^m$ assumed to be drawn from i.i.d. random variables with unknown distribution $p(\boldsymbol{x}, \boldsymbol{y})$ defined over $\mathcal{X} \times \mathcal{Y}$, the goal of learning is to find a classifier h from a given hypothesis space $\mathcal{H}$ such that the expected risk is minimal, i.e.

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} R_{\exp}(\boldsymbol{h}) := \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{y} \in \mathcal{Y}} p(\boldsymbol{x}, \boldsymbol{y}) \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \,. \tag{1}$$

The direct minimization of $R_{\exp}(\boldsymbol{h})$ is impossible due to the unknown distribution $p(\boldsymbol{x}, \boldsymbol{y})$. A discriminative approach to learning is based on replacing the unknown distribution $p(\boldsymbol{x}, \boldsymbol{y})$ by its empirical distribution $p_{\text{emp}}(\boldsymbol{x}, \boldsymbol{y}) = \frac{1}{m} \sum_{i=1}^{m} [\![\boldsymbol{x} = \boldsymbol{x}^i \wedge \boldsymbol{y} = \boldsymbol{y}^i]\!]$ which leads to the emprical risk minimization

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} R_{\text{emp}}(\boldsymbol{h}) := \frac{1}{m} \sum_{i=1}^{m} \ell(\boldsymbol{y}^i, \boldsymbol{h}(\boldsymbol{x}^i)) \,. \tag{2}$$

The discriminative learning is implemented in the structured output SVM [4]. In this case, the hypothesis space $\mathcal{H}$ is a set of linear classifiers

$$\boldsymbol{h}(\boldsymbol{x}; \boldsymbol{w}) = \operatorname*{argmax}_{\boldsymbol{y} \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \tag{3}$$

with parameter vector $\boldsymbol{w}$ from a ball $\mathcal{W} = \{\boldsymbol{w} \in \Re^n \mid \|\boldsymbol{w}\| \leq r\}$ of fixed radius r. The input-output feature map $\boldsymbol{\Psi} \colon \mathcal{X} \times \mathcal{Y} \to \Re^n$ is assumed to be fixed and only the parameter vector $\boldsymbol{w} \in \mathcal{W}$ is learned from examples. The minimization of the empirical risk w.r.t. the class of linear classifiers is known to be a hard problem for most loss functions used in the structured output classification. Hence, the original loss $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \Re$ is replaced by a convex surrogate loss function which can efficiently optimized. For example, the margin-rescaling loss

$$\hat{\ell}(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}; \boldsymbol{w})) = \max_{\boldsymbol{y}' \in \mathcal{Y}} \left[\ell(\boldsymbol{y}, \boldsymbol{y}') + \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}') \right] - \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \tag{4}$$

is among most frequently used in structured output learning due existence of algorithms for its efficient evaluation in a wide range of models. The structured output SVM learning then leads to a convex optimization problem

$$\boldsymbol{w}^* = \operatorname*{argmin}_{\boldsymbol{w} \in \mathcal{W}} R_{\text{svm}}(\boldsymbol{w}) := \frac{1}{m} \sum_{i=1}^{m} \hat{\ell}(\boldsymbol{y}^i, \boldsymbol{h}(\boldsymbol{x}^i, \boldsymbol{w})) \,,$$

which can be solved efficiently.

3 Discriminative learning from partially annotated examples

3.1 Generative model of partially annotated examples

Let $\boldsymbol{x} \in \mathcal{X}$, $\boldsymbol{y} = (y_t \in \mathcal{Y}_t \mid t \in \mathcal{T}) \in \mathcal{Y}$ and $\ell(\boldsymbol{y}, \boldsymbol{y}')$ be defined as in the fully supervised case described in Section 2. In addition to this, let $\boldsymbol{z} = (z_t \in \{0,1\} \mid t \in \mathcal{T}) \in \mathcal{Z}$ be a vector of binary variables, which we call the *annotation mask*, indicating which labels in the training set are annotated. In particular, $z_t = 1$ means that y_t is annotated while $z_t = 0$ means that the label y_t is not annotated. We assume that $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ are generated according to the joint distribution

$$p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z}) = p(\boldsymbol{x}) p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{z} \mid \boldsymbol{x}) \,. \tag{5}$$

Let $\boldsymbol{a} = (a_t \in \mathcal{Y}_t \cup \{?\} \mid t \in \mathcal{T}) \in \mathcal{A}$ be an annotation created from the labeling $\boldsymbol{y} \in \mathcal{Y}$ and the annotation mask $\boldsymbol{z} \in \mathcal{Z}$ by the function $\boldsymbol{\alpha} \colon \mathcal{Y} \times \mathcal{Z} \to \mathcal{A}$ defined as

$$\boldsymbol{a} = \boldsymbol{\alpha}(\boldsymbol{y}, \boldsymbol{z}) \qquad \text{where} \qquad a_t = \begin{cases} y_t & \text{if} \quad z_t = 1 \,, \\ ? & \text{if} \quad z_t = 0 \,. \end{cases}$$

That is, the annotation $\boldsymbol{a} = \boldsymbol{\alpha}(\boldsymbol{y}, \boldsymbol{z})$ is created from $\boldsymbol{y}$ and $\boldsymbol{z}$ by copying labels of the objects assigned for annotation $\{t \in \mathcal{T} \mid z_t = 1\}$ while the remaining objects $\{t \in \mathcal{T} \mid z_t = 0\}$ get the special label "?". Because the annotation $\boldsymbol{a}$ is a deterministic function of random variables $\boldsymbol{y}$ and $\boldsymbol{z}$ it is also a random variable with distribution

$$p(\boldsymbol{a} \mid \boldsymbol{y}, \boldsymbol{z}) = [\![\boldsymbol{a} = \boldsymbol{\alpha}(\boldsymbol{y}, \boldsymbol{z})]\!] \,. \tag{6}$$

Using (5) and (6) it follows that

$$p(\boldsymbol{x}, \boldsymbol{a}) = p(\boldsymbol{x}) \sum_{\boldsymbol{y} \in \mathcal{Y}} \sum_{\boldsymbol{z} \in \mathcal{Z}} p(\boldsymbol{y} \mid \boldsymbol{x}) p(\boldsymbol{z} \mid \boldsymbol{x}) [\![\boldsymbol{a} = \boldsymbol{\alpha}(\boldsymbol{y}, \boldsymbol{z})]\!] \,.$$

which describes a random process generating a set of partially annotated examples $\{(\boldsymbol{x}^1, \boldsymbol{a}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{a}^m)\}$.

In the formulation of the learning problem we will need the distribution $p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{a})$. From (5) and (6) we can derive

$$p(\boldsymbol{y} \,|\, \boldsymbol{x}, \boldsymbol{a}) = \frac{p(\boldsymbol{y} \,|\, \boldsymbol{x}) c(\boldsymbol{y}, \boldsymbol{a})}{\sum_{\boldsymbol{y}' \in \mathcal{Y}} p(\boldsymbol{y}' \,|\, \boldsymbol{x}) c(\boldsymbol{y}', \boldsymbol{a})}$$

where

$$c(\boldsymbol{y}, \boldsymbol{a}) = \prod_{t \in \mathcal{T}} [\![y_t = a_t \vee a_t = ?]\!] \,.$$

3.2 Interpretation of the proposed generative model

We can interpret the generative model $p(\boldsymbol{x})p(\boldsymbol{y} \mid \boldsymbol{x})$ in the usual way [Vapnik-Nature2005]. The distribution $p(\boldsymbol{x})$ describes how the nature generates observations $\boldsymbol{x}$. The set of labels $\mathcal{Y}$ is defined artificially by a designer according to the application at hand. The distribution $p(\boldsymbol{y} \mid \boldsymbol{x})$ describes how an annotator assigns labels to an observation $\boldsymbol{x}$.

We have augmented the standard model by introducing the annotation mask $\boldsymbol{z} = (z_t \in \{0, 1\} \mid t \in \mathcal{T}) \in \mathcal{Z}$ distributed according to $p(\boldsymbol{z} \mid \boldsymbol{x})$. We will call $p(\boldsymbol{z} \mid \boldsymbol{x})$ the *annotation scheme*. The distribution $p(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ given by (5) generates a triplet $(\boldsymbol{x}, \boldsymbol{y}, \boldsymbol{z})$ which induces a training example pair $(\boldsymbol{x}, \boldsymbol{a})$ composed of the observation $\boldsymbol{x}$ and its partial annotation $\boldsymbol{a} = \boldsymbol{\alpha}(\boldsymbol{y}, \boldsymbol{z})$. The annotation $\boldsymbol{a} = \boldsymbol{\alpha}(\boldsymbol{y}, \boldsymbol{z})$ is created by copying labels of the annotated objects $\{t \in \mathcal{T} \mid z_t = 1\}$ from $\boldsymbol{y}$ while the remaining objects $\{t \in \mathcal{T} \mid z_t = 0\}$ get the special label "?". Intuitively, the value of a conditional marginal probability $p(z_t = 1 \mid \boldsymbol{x})$ (or $p(z_t = 0 \mid \boldsymbol{x})$) can be interpreted as easiness (or hardness) of annotating the object $t \in \mathcal{T}$. The conditional independence $p(\boldsymbol{y}, \boldsymbol{z} \,|\, \boldsymbol{x}) = p(\boldsymbol{y} \,|\, \boldsymbol{x})p(\boldsymbol{z} \,|\, \boldsymbol{x})$ assumed in the model (5) follows from the fact that both the labels $\boldsymbol{y}$ and the annotation mask $\boldsymbol{z}$ are determined by the annotator using the information from the observation $\boldsymbol{x}$.

Later in this report we will propose a learning algorithm applicable in the case when the annotation scheme $p(\boldsymbol{z} \mid \boldsymbol{x})$ is i) known and ii) proper in the following sense:

Definition 1 *The annotation scheme defined by distribution $p(\boldsymbol{z} \mid \boldsymbol{x})$ is called proper if all conditional marginal probabilities are non-zero, i.e. $p(z_t = 1 \,|\, \boldsymbol{x}) > 0, \forall t \in \mathcal{T}$, $\boldsymbol{x} \in \mathcal{X}$*

The proper annotation scheme implies that every object has a chance to be annotated regardless the input $\boldsymbol{x}$.

Is it a realistic assumption to require the annotation scheme to be known and proper? Let us consider a class of applications of the structured output learning where these assumptions are in place. In particular, the computer vision applications using the deformable part models learned from annotated images belong to this class. The learned defomable part models are commonly used for detection, segmentation and tracking of complex objects in images. A prototypical application considered in this report is a face landmark detector learned from examples. The landmarks are well discriminative features defined on the human face like the corners of eyes, the nose and the corners of mouth. The structured output classifier is used to estimate positions of the landmarks in a given image. The so far used fully supervised

structured output learning requires a large training set of face images along with manual annotation of all landmark positions in each image.

In the fully supervised case, the annotator is asked to mark positions of all landmarks which corresponds to the annotation scheme $p(z_t \mid \boldsymbol{x}) = 1$, $\forall t \in \mathcal{T}$. However, we can instruct the annotator to mark only a subset of landmarks using, for example, the following annotation scheme:

- In every image the annotator marks position of the nose tip $y_1 \in \mathcal{Y}_1$.
- In each even image the annotator marks only the positions of landmarks on the left part of the face $(y_t \in \mathcal{Y}_t \mid t \in \mathcal{T}_{\text{left}})$.
- In each odd image the annotator marks only the positions of landmarks on the right part of the face $(y_t \in \mathcal{Y}_t \mid t \in \mathcal{T}_{\text{right}})$.

Provided the annotator follows these instructions and the images are presented in a random order (which we can easily assure by randomly reshuffling the images before annotation) implies that

$$p(z_1 \mid \boldsymbol{x}) = 1 \quad \text{and} \quad p(z_t \mid \boldsymbol{x}) = \frac{1}{2}\,, \quad t \in \mathcal{T}_{\text{left}} \cup \mathcal{T}_{\text{right}}\,.$$

By using this procedure the annotator marks approximately a half of the landmarks and we known the exact annotation scheme.

3.3 Formulation of the learning problem

In this section we formulate the problem of learning the structured output classifier from partially annotated examples.

Let us assume that we are given a loss function $\ell\colon \mathcal{Y} \times \mathcal{Y} \to \Re$ and a training multi-set of examples $\{(\boldsymbol{x}^1, \boldsymbol{a}^1), \ldots, (\boldsymbol{x}^m, \boldsymbol{a}^m)\} \in (\mathcal{X} \times \mathcal{A})^m$ drawn from i.i.d. random variables with an unknown distribution $p(\boldsymbol{x}, \boldsymbol{a})$ which belongs to the class of distribution defined in Section 3.1. The goal of learning is to find a classifier $\boldsymbol{h}\colon \mathcal{X} \to \mathcal{Y}$ from a given hypothesis space $\mathcal{H}$ such that the expected risk is minimal, i.e.

$$\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} R_{\exp}(\boldsymbol{h}) := \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{y} \in \mathcal{Y}} \sum_{\boldsymbol{a} \in \mathcal{A}} p(\boldsymbol{x}, \boldsymbol{a})\, p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a})\, \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x}))\,. \tag{7}$$

The expected risk $R_{\exp}(\boldsymbol{h})$ in the partial learning problem (7) coincides with the expected risk in the formulation of the fully supervised learning (1). The only difference is that in the partial learning setting we have to marginalize over all possible annotations.

Our goal is to use the discriminative approach to solve the learning problem (7), i.e. we want to avoid modeling the distribution functions appearing in the problem (7). To this end, we define a partial loss which can be evaluated provided the annotation scheme $p(\boldsymbol{z} \mid \boldsymbol{x})$ is known. We show that minimization of the partial loss is equivalent to the task (7).

In the following we will assume that the loss function $\ell(\boldsymbol{y}, \boldsymbol{y}')$ is *additively decomposable over the objects*, i.e.

$$\ell(\boldsymbol{y}, \boldsymbol{y}') = \sum_{t \in \mathcal{T}} \ell_t(y_t, y'_t) \,,$$

where $\ell_t \colon \mathcal{Y}_t \times \mathcal{Y}_t \to \Re$ are partial loss functions for individual objects $\mathcal{T}$.

Definition 2 *Let $\ell \colon \mathcal{Y} \times \mathcal{Y} \to \Re$ be an additively decomposable loss function and $p(\boldsymbol{z} \mid \boldsymbol{x})$ a proper annotation scheme. Then, the partial loss $\ell^{\mathrm{P}} \colon \mathcal{X} \times \mathcal{A} \times \mathcal{Y} \to \Re$ associated with $\ell(\boldsymbol{y}, \boldsymbol{y}')$ and $p(\boldsymbol{z} \mid \boldsymbol{x})$ is defined as*

$$\ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}) = \sum_{t \in \mathcal{T}} [\![a_t \neq ?]\!] \frac{\ell_t(a_t, y_t)}{p(z_t = 1 \mid \boldsymbol{x})}$$

Now we can define a partial risk minimization problem

$$\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{h} \in \mathcal{H}} R_{\exp}(\boldsymbol{w}) := \sum_{\boldsymbol{x} \in \mathcal{X}} \sum_{\boldsymbol{a} \in \mathcal{A}} p(\boldsymbol{x}, \boldsymbol{a}) \ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{h}(\boldsymbol{x})) \,, \tag{8}$$

which is equivalent to the learning problem (7) because the objective functions of both problems have the same value for any $\boldsymbol{h}$ (hence we denote both objective function by $R_{\exp}(\boldsymbol{w})$) due to the following theorem.

Theorem 1 *The equality*

$$\sum_{\boldsymbol{y} \in \mathcal{Y}} \sum_{\boldsymbol{a} \in \mathcal{A}} p(\boldsymbol{a} \,|\, \boldsymbol{x}) p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}) \ell(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \sum_{\mathcal{A} \in \mathcal{P}(\mathcal{Y})} p(\boldsymbol{a} \mid \boldsymbol{x}) \ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{h}(\boldsymbol{x}))$$

holds true for any $\boldsymbol{x} \in \mathcal{X}$ and $\boldsymbol{h} \in \mathcal{H}$.

PROOF: We can write

$$
\begin{aligned}
& \sum_{\boldsymbol{a}\in\mathcal{A}} p(\boldsymbol{a} \mid \boldsymbol{x})\ell^{\mathrm{p}}(\boldsymbol{x},\boldsymbol{a},\boldsymbol{h}(\boldsymbol{x})) \\
= & \sum_{\boldsymbol{a}\in\mathcal{A}}\sum_{\boldsymbol{y}\in\mathcal{Y}}\sum_{\boldsymbol{z}\in\mathcal{Z}} p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{z} \mid \boldsymbol{x})[\![\mathcal{A}=\boldsymbol{a}(\boldsymbol{y},\boldsymbol{z})]\!]\sum_{t\in\mathcal{T}}[\![a_t \neq ?]\!]\frac{\ell_t(a_t,h_t(\boldsymbol{x}))}{p(z_t=1 \mid \boldsymbol{x})} \\
= & \sum_{\boldsymbol{y}\in\mathcal{Y}}\sum_{\boldsymbol{z}\in\mathcal{Z}} p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{z} \mid \boldsymbol{x})\sum_{\boldsymbol{a}\in\mathcal{A}}[\![\mathcal{A}=\boldsymbol{a}(\boldsymbol{y},\boldsymbol{z})]\!]\sum_{t\in\mathcal{T}}[\![a_t \neq ?]\!]\frac{\ell_t(a_t,h_t(\boldsymbol{x}))}{p(z_t=1 \mid \boldsymbol{x})} \\
= & \sum_{\boldsymbol{y}\in\mathcal{Y}}\sum_{\boldsymbol{z}\in\mathcal{Z}} p(\boldsymbol{y} \mid \boldsymbol{x})p(\boldsymbol{z} \mid \boldsymbol{x})\sum_{t\in\mathcal{T}} z_t\frac{\ell_t(y_t,h_t(\boldsymbol{x}))}{p(z_t=1 \mid \boldsymbol{x})} \\
= & \sum_{\boldsymbol{y}\in\mathcal{Y}} p(\boldsymbol{y} \mid \boldsymbol{x})\sum_{t\in\mathcal{T}}\sum_{z_t\in\{0,1\}} z_t\ p(z_t \mid \boldsymbol{x})\frac{\ell_t(y_t,h_t(\boldsymbol{x}))}{p(z_t=1 \mid \boldsymbol{x})} \\
= & \sum_{\boldsymbol{y}\in\mathcal{Y}} p(\boldsymbol{y} \mid \boldsymbol{x})\sum_{t\in\mathcal{T}}\ell_t(y_t,h_t(\boldsymbol{x})) \\
= & \sum_{\boldsymbol{y}\in\mathcal{Y}}\sum_{\boldsymbol{a}\in\mathcal{A}} p(\boldsymbol{a} \mid \boldsymbol{x})p(\boldsymbol{y} \mid \boldsymbol{x},\boldsymbol{a})\ell(\boldsymbol{y},\boldsymbol{h}(\boldsymbol{x}))
\end{aligned}
$$

■

Following the discriminative approach to learning we replace $p(\boldsymbol{x},\boldsymbol{a})$ in the partial risk minimization problem (8) by its empirical counterpart $p_{\mathrm{emp}}(\boldsymbol{x},\boldsymbol{a}) = \frac{1}{m}\sum_{i=1}^{m}[\![\boldsymbol{x}=\boldsymbol{x}^i \wedge \boldsymbol{y}=\boldsymbol{y}^i]\!]$ which yields

$$
\boldsymbol{h}^* \in \operatorname*{argmin}_{\boldsymbol{h}\in\mathcal{H}} R^{\mathrm{p}}_{\mathrm{emp}}(\boldsymbol{h}) = \frac{1}{m}\sum_{i=1}^{m}\ell^{\mathrm{p}}(\boldsymbol{x}^i,\boldsymbol{a}^i,\boldsymbol{h}(\boldsymbol{x}^i))\,. \tag{9}
$$

4 Learning of linear classifiers from partially annotated examples

In this section we will discuss an instance of the discriminative partial learning (9) for the class of linear classifiers (3), i.e. the learning problem (9) becomes

$$
\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{w}\in\Re^n} R^{\mathrm{p}}_{\mathrm{emp}}(\boldsymbol{w}) = \frac{1}{m}\sum_{i=1}^{m}\ell^{\mathrm{p}}(\boldsymbol{x}^i,\boldsymbol{a}^i,\boldsymbol{h}(\boldsymbol{x}^i,\boldsymbol{w}))\,. \tag{10}
$$

The optimization problem (10) is hard for most useful instances of the loss function ℓ^{p}. By adopting the idea of margin-rescaling loss (4) we can approx-

imate the partial loss $\ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{h}(\boldsymbol{x}))$ as follows

$$
\begin{aligned}
\ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{h}(\boldsymbol{x})) &= \max_{\boldsymbol{y} \in \mathcal{Y}} \ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}) [\![\max_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}') \rangle - \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \leq 0]\!] \\
&\leq \max_{\boldsymbol{y} \in \mathcal{Y}} \max \left\{ 0, \ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}) - \max_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}') \rangle + \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \right\} \\
&= \underbrace{\max_{\boldsymbol{y} \in \mathcal{Y}} \left[\ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}) + \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \right]}_{\hat{\ell}^{\mathrm{P}}_{\mathrm{vex}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{w})} \underbrace{- \max_{\boldsymbol{y}' \in \mathcal{Y}} \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}') \rangle}_{\hat{\ell}^{\mathrm{P}}_{\mathrm{cave}}(\boldsymbol{x}, \boldsymbol{w})} \\
&= \hat{\ell}^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{w})
\end{aligned}
$$

where the transition from the second to the third line requires that the loss is non-negative, i.e. $\ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}) \geq 0$, $\forall \boldsymbol{x}, \boldsymbol{a}, \boldsymbol{y}$. The obtained surrogate loss $\hat{\ell}^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{w})$, which we call margin-rescaling partial loss, is an upper bound of the original partial loss $\ell^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{w})$. By sub-substituting $\hat{\ell}^{\mathrm{P}}(\boldsymbol{x}, \boldsymbol{a}, \boldsymbol{w})$ to (10) and constraining the parameters space to the ball $\mathcal{W}$ like in the structured SVMs, we get the following optimization problem which we denote as the *Partial-SO-SVM problem*:

$$
\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{w} \in \mathcal{W}} F(\boldsymbol{w}) := \frac{1}{m} \left[\sum_{i=1}^{m} \hat{\ell}^{\mathrm{P}}_{\mathrm{vex}}(\boldsymbol{x}^i, \boldsymbol{a}^i, \boldsymbol{w}) + \sum_{i=1}^{m} \hat{\ell}^{\mathrm{P}}_{\mathrm{cave}}(\boldsymbol{x}^i, \boldsymbol{w}) \right] . \qquad (11)
$$

The Partial-SO-SVM problem (11) is a sum of convex and concave function. No feasible algorithm finding a global optimum of (11) has been proposed so far. The most popular approach for finding a local optima of functions in the form of (11) is the Convex Concave Procedure (CCCP) [1]. The CCCP iteratively solves a convex relaxation of (11) obtained by replacing the concave part of $\hat{R}^{\mathrm{P}}_{\mathrm{emp}}(\boldsymbol{w})$ by its first order Tailor expansion computed at the current solution. The algorithm alternates two steps. In the first step, the linear approximation of the concave part is computed which amounts to classifying the training examples using the linear classifier with current parameter vector. In the second step, the relaxed objective is optimized which amounts to solving a task resembling the supervised learning problem and thus existing solvers can be recycled. The main advantage of the CCCP is its simplicity. The main disadvantage is that the result is sensitive to the initialization and that no certificate of optimality is provided. Additional complication stems from the iterative nature of the CCCP as each iteration requires solving the supervised learning problem its complexity is not negligible for real-life data.

4.1 Solving the Partial-SO-SVM problem using additional prior knowledge

Here we propose a heuristic method solves a surrogate problem approximating the Partial-SO-SVM problem (10). Our method requires additional prior knowledge. In particular, let us assume we have a parametric distribution $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$, $\boldsymbol{\theta} \in \Theta$, approximating the true posterior probability of the labeling $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a})$. It is important to note that estimation of $p(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a})$ can be much easier than estimation of $p(\boldsymbol{y} \mid \boldsymbol{x})$.

We propose to approximate the concave function $\hat{\ell}^{\mathrm{p}}_{\mathrm{cave}}(\boldsymbol{x}, \boldsymbol{w})$ by a linear (and thus convex) term

$$\hat{\ell}^{\mathrm{p}}_{\mathrm{lin}}(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) = -\sum_{\boldsymbol{y} \in \mathcal{Y}} \hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle \,, \tag{12}$$

i.e. we have replaced the maximization w.r.t. the label $\boldsymbol{y}$ by the expectation w.r.t. posterior of $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$. Substituting (12) to (11) and adding minimization w.r.t. $\boldsymbol{\theta}$ we get a surrogate Partial-SO-SVM problem

$$\boldsymbol{w}^* \in \operatorname*{argmin}_{\boldsymbol{\theta} \in \Theta, \boldsymbol{w} \in \mathcal{W}} \hat{F}(\boldsymbol{w}, \boldsymbol{\theta}) := \frac{1}{m} \left[\sum_{i=1}^{m} \hat{\ell}^{\mathrm{p}}_{\mathrm{vex}}(\boldsymbol{x}^i, \boldsymbol{a}^i, \boldsymbol{w}) + \sum_{i=1}^{m} \hat{\ell}^{\mathrm{p}}_{\mathrm{lin}}(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \right] . \tag{13}$$

The following arguments support the surrogate Partial-SO-SVM problem (13):

- In the case of fully annotated training examples (i.e. when $a^i_t \neq ?$, $\forall i \in 1, \ldots, m, t \in T$) the problem (13) becomes the standard supervised SO-SVM with margin-rescaling loss function.
- Regardless of the form of the parametric distribution $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ the function $\hat{F}(\boldsymbol{w}, \boldsymbol{\theta})$ is an upper bound of the true empirical risk $R^{\mathrm{p}}_{\mathrm{emp}}(\boldsymbol{w})$ we want to minimize.
- In the case the parametric distribution $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ has the most generic form, i.e. it becomes a non-parametrix distribution different for each example, the surrogate learning problem (13) reduces to the Partial-SO-SVM problem (11).

The surrogate Partial-SO-SVM problem (13) requires minimization of the objective $\hat{F}(\boldsymbol{w}, \boldsymbol{\theta})$ w.r.t. the model parameters $\boldsymbol{w}$ and the distribution parameters $\boldsymbol{\theta}$. A straightforward method to optimize (13) is to use a block coordinate descent, i.e. we can alternate minimization w.r.t $\boldsymbol{w}$ while $\boldsymbol{\theta}$ is fixed and vice-versa. The minimization w.r.t. $\boldsymbol{w}$ reduces to a convex problem resembling the supervised SO-SVM hence existing solvers can be used. A

suitable strategy for minimization of $\hat{F}(\boldsymbol{w}, \boldsymbol{\theta})$ w.r.t $\boldsymbol{\theta}$ depends on the form of $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$. As mentioned above, one extreme case occurs if $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ is non-parametric which reduces the block-coordinate descend to the CCCP algorithm for the Partial-SO-SVM. Another extreme case occurs if $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ is set to the uniform distribution. In this case, the solution is obtained in a single iteration. This case resembles the partial loss for the flat classification proposed in [CourTaskar]. In the experiments presented in this report the parameter set Θ is finite with small cardinality hence an exhaustive search is possible. This means that the partial learning will reduce to a series of problems resembling the supervised SO-SVM.

Solving the surrogate learning problem (13) requires evaluation of the linear term $\hat{\ell}^{\mathrm{p}}_{\mathrm{lin}}(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ given by (12). This term can be evaluated efficiently for a generic structured output linear classifier based on the Pairwise Markov Network (PMN) as we show next.

Let $\mathcal{T}$ be a set of nodes and let $\mathcal{E} \subseteq \binom{\mathcal{T}}{2}$ be a set of edges defining a graph $(\mathcal{T}, \mathcal{E})$. The PMN classifier is an instance of the linear classifier (3) with the input-output feature map $\Psi \colon \mathcal{X} \times \mathcal{Y} \to \Re^n$ defined as

$$\Psi(x, y) = \sum_{t \in \mathcal{T}} \Psi_t(x, y_t) + \sum_{tt' \in \mathcal{E}} \Psi_{tt'}(x, y_t, y_{t'}) \,. \tag{14}$$

Evaluation of the linear term $\hat{\ell}^{\mathrm{p}}_{\mathrm{lin}}(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ for the linear classifier with the input-output map (14) amounts to computation of

$$\hat{\ell}^{\mathrm{p}}_{\mathrm{lin}}(\boldsymbol{w}, \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) = -\sum_{\boldsymbol{y} \in \mathcal{Y}} \hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle = \langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \rangle \,,$$

where

$$\begin{aligned} \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \;\; = \;\; & \sum_{t \in \mathcal{T}} \sum_{y_t \in \mathcal{Y}_t} \hat{p}_t(y_t \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \Psi_t(\boldsymbol{x}, y_t) \\ & + \sum_{tt' \in \mathcal{E}} \sum_{y_t \in \mathcal{Y}_t} \sum_{y_{t'} \in \mathcal{Y}_{t'}} \hat{p}_{tt'}(y_t, y_{t'} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) \Psi_{tt'}(\boldsymbol{x}, y_t, y_{t'}) \,, \end{aligned}$$

and $\hat{p}_t(y_t \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$, $\hat{p}_{tt'}(y_t, y_{t'} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ are object and edge marginal probabilities derived from $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$. Given the marginal probabilities, the computation of the vector $\boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ requires $\mathcal{O}(n|\mathcal{T}||\mathcal{Y}| + n|\mathcal{E}||\mathcal{Y}|^2)$ operations. Note that it holds regardless the structure of the graph $(\mathcal{T}, \mathcal{E})$.

5 Experiments

5.1 Detector of Face Landmarks

In this section we consider problem of learning face landmark detector from examples. We follow the approach of [5] where the landmark detection is posed as an instance of the structured output linear classifier (3). The classifier input $x \in \mathcal{X}$ is an image 40×40 pixels large which contains a face. The classifier outputs $\boldsymbol{y} = (y_t \in \mathcal{Y}_t \mid t \in \mathcal{T} = \{0, 1, \ldots, 7\}) \in \mathcal{Y}$ where $\mathcal{Y}_t \subset 40 \times 40$ is a set of admissible 2D coordinates for t-th landmark. There are eight landmarks including the corners of the eyes, the corners of the mouth, tip of the nose and the center of the face. The scoring function $\langle \boldsymbol{w}, \boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y}) \rangle$ of the classifier (3) is composed of the appearance model and the deformation cost. The appearance model evaluates a match between the input image $\boldsymbol{x}$ and the landmark templates put at positions $\boldsymbol{y}$. The deformation cost evaluates likeliness of the particular landmark configuration $\boldsymbol{y}$ and this cost decomposes to a set of pair wise terms defined over edges of an acyclic graph. The map $\boldsymbol{\Psi}(\boldsymbol{x}, \boldsymbol{y})$ is as a column-wise concatenation of local feature descriptors of individual landmarks and parameters of the deformation cost. We use a variant of Local Binary Patterns as the feature descriptor. Evaluation of the classifier (3) leads to solving an instance of the dynamic programing. The loss function measures the mean deviation between the ground truth landmark positions $\boldsymbol{y}$ and their estimate $\boldsymbol{y}'$, i.e. $\ell(\boldsymbol{y}, \boldsymbol{y}') = \kappa(\boldsymbol{y}) \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \|y_t - y'_t\|$, where $\kappa(\boldsymbol{y})$ is a normalization constant ensuring that the loss is scale invariant.

We use the same data and the testing protocol as in [5]. The difference is that while [5] learns parameters of the landmark detector from fully annotated examples (i.e. position of each landmark is marked in the training image) here we learn the detector from only partially annotated examples. We use the annotation scheme $p(\boldsymbol{z} \mid \boldsymbol{x})$ described in the example presented in Section 3.2, i.e. the annotator is requested to mark in each image the nose position and only half of the pair landmarks. Hence this annotation scheme requires approximately half of the effort as compared to the full annotation.

The distribution $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ is modeled as follows. Let us assume that $(\boldsymbol{x}, \boldsymbol{a})$ is a training image along with the positions of the landmarks on the right hand side of the face. We assume that

$$\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) = \prod_{t \in \mathcal{T}} \hat{p}(y_t \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$$

and that the $\hat{p}(y_t \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) = [\![a_t = y_t]\!]$, $t \in \mathcal{T}_{\text{right}}$, i.e. distribution of positions for each landmark is modeled independently and the annotated

	θ	Training Error	Test error
Fully annotated examples	–	4.097	5.659
Partially annotated examples	0.0005	6.628	7.636
	0.005	7.350	
	0.05	7.238	
	0.5	8.055	

Table 1: Errors for the face landmark detector learned from the fully annoated examples and the partially annotated examples. In the case of the partial learning the training errors are reported for different distribution parameter θ.

landmarks have the Dirac distribution centered at the annotated position. In the modeling of the distribution of the unannotated landmarks we use the fact that the pair organs on the face are vertically symmetric, i.e. each left landmark has its right counterpart up to the nose which is, however, always annotated. We set the distribution of the unannotated landmarks to be

$$\hat{p}(y_t \mid \boldsymbol{x}, \boldsymbol{a}; \theta) \approx \exp\left(- \frac{\|\overline{\boldsymbol{I}}(\boldsymbol{x}, y_t) - \boldsymbol{I}(\boldsymbol{x}, y_{t'})\|^2}{\theta} \right),$$

where $t \in \mathcal{T}_{\text{left}}$ is an unannotated landmark and $t' \in \mathcal{T}_{\text{right}}$ is its corresponding annotated counterpart. For example, t and t' can be the left and the right mouth corners. The vector $\boldsymbol{I}(\boldsymbol{x}, y_t)$ is a column wise representation of intensity values in a sub-window of the image $\boldsymbol{x}$ centered around y_t. The vector $\overline{\boldsymbol{I}}(\boldsymbol{x}, y_t)$ is constructed likewise, however, the sub-window is vertically mirrored. The distributions have a single scalar parameter θ. We find the optimal value of $\boldsymbol{\theta}$ by minimizing over a finite set of values $\Theta = \{0.0005, 0.005, \ldots, 0.5\}$. Hence the partial learning is transformed to $|\Theta|$ supervised SO-SVM learning problems and selecting the one with the lowest objective function.

The overall results are presented in Table 1. It is seen that the partial learning yields slightly worse results than learning from fully annotated examples, however, it the minor lost in accuracy is compensated with saving half of the annotators time.

5.2 Image segmentation

In this section we consider experiment with the semantic image segmentation. Given an input color image, the goal is to assign each image pixel to one semantic class. The automatic assignment pixels to classes is treated as

the structured output classification problem. The classifier here is a pairwise Markov Network with super-modular potential functions (SM-PMN). The supervised SO-SVM learning is in the case of SM-PMN polynomially tractable. We use exactly the same setting of the SM-PMN classifier, the database and the testing protocol as in the paper [2]. The main difference is that [2] learn the parameters of the SM-PMN from fully annotated examples while we consider only partially annotated ones.

The full annotation in this case means that the annotator has to manually assign a label to each pixel in the image. In particular, we considered a subset of the MSCR database containing images of cows in a nature scene. We had three semantic classes: *grass, sky, cow*. A rough manual annotation of interior parts of the three segments is easy and it can be done in a few seconds. On the other hand, a precise annotation of the boundaries between the segments (e.g. delineated exactly the cow on the grass) is a tedious work which can take tens of minutes.

We took the pixel precise annotations of the images and use them to generate partial annotations with gradually decreasing precision. In particular, each boundary pixel in the precision annotation was used as a center of a circle whose inner pixels were assigned a special label ? (i.e. unlabeled pixel). The radius of the circle was gradually increased from $r = 0$ (the original precise annotation with no unlabeled pixels) to $r = 30$ (the boundary band has width 30 pixels). The figure 2(a) shows an example of a training image and its different partial annotation created by gradually increasing the radius r.

In order to apply the partial learning algorithm proposed in this work we need to specify the annotation scheme $p(z_t \mid \boldsymbol{x})$ and the distribution $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ required for solving the surrogate Partial-SO-SVM. The annotation scheme is set to be $p(z_t \mid \boldsymbol{x}) = p(z_{t'} \mid \boldsymbol{x})$, $\forall t, t'$, i.e. the difficulty of labeling a pixels does not depend on its location in the image. This assumption is clearly not satisfied as the boundary pixels are less likely to be labeled. To approximate $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ we assume that the distribution decompose to a product

$$\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta}) = \prod_{t \in \mathcal{T}} N(x_t; \mu_{y_t}, \sigma_{y_t})$$

where the parameters $(\mu_{y_t}, \sigma_{y_t})$, $y_t \in \{grass, sky, cow\}$ of the Gaussians are estimated from the annotated pixels $\boldsymbol{a}$ of the training image. It is seen that both in modeling the annotation scheme $p(\boldsymbol{z} \mid \boldsymbol{x})$ and the distribution $\hat{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{a}; \boldsymbol{\theta})$ quite crude approximations were used yet the results are surprisingly good. The results are summarized in Table 2 and Figure 1 which shows the training and testing errors for the SM-PMN classifier trained from the fully ($r = 0$) and partially annotated examples with different amount of unlabeled

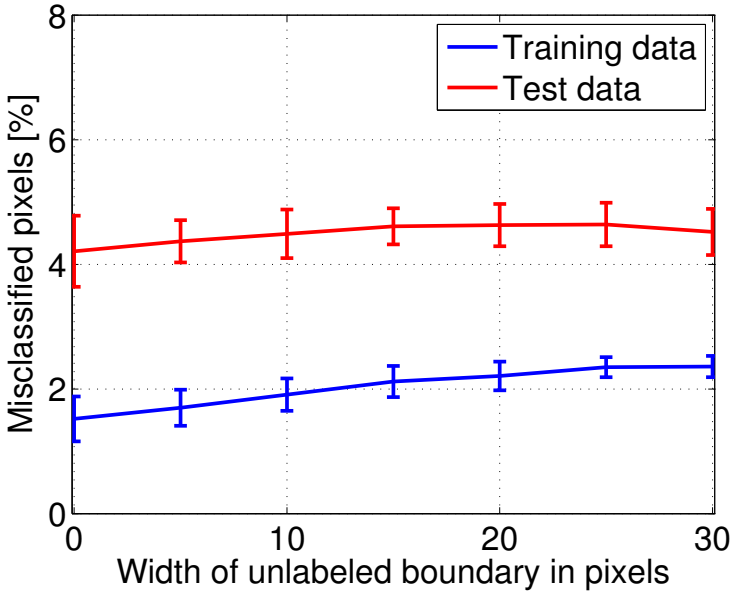
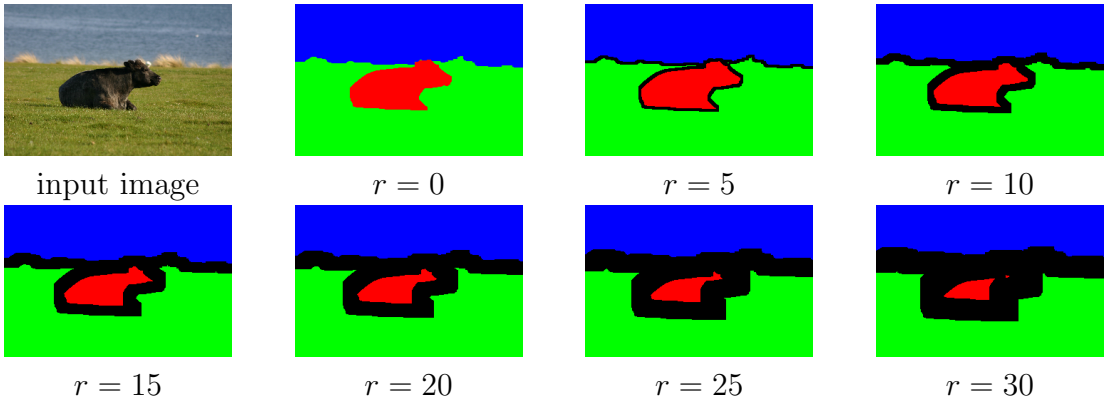


Figure 1: results

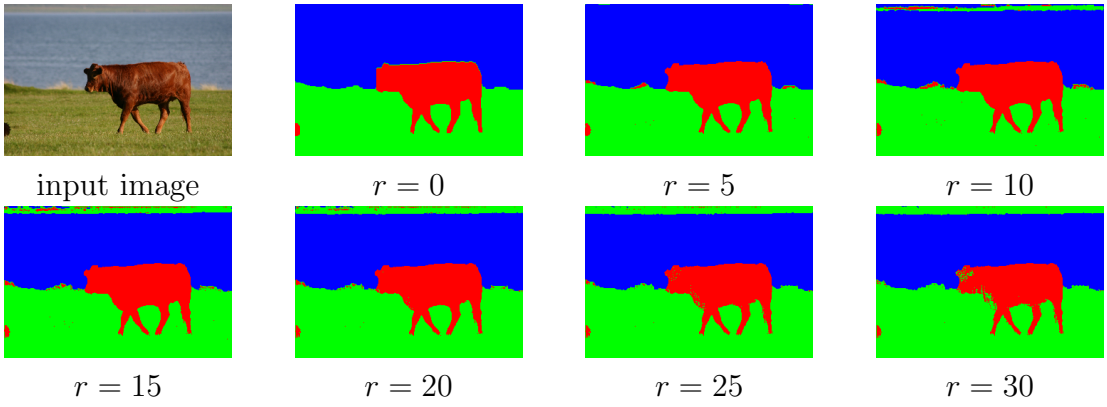
pixels. It is seen that the error grows with increasing amount of unlabeled pixels as expected, however, the slope of the error curves is small. It is seen on test examples shown in Figures 2(b) and 2(c) that the qualitatively of the segmentations obtained from the fully supervised learning and the partial learning is comparable.

6 Conclusions

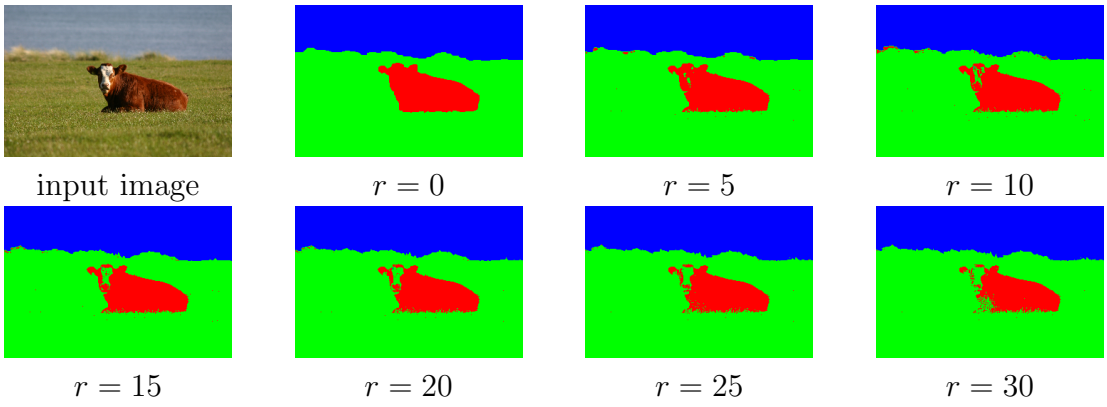
We have provided a statistical formulation of the problem of learning the structured output classifiers from partially annotated examples. We formulated learning as an instance of the expected risk minimization. We showed that the minimization of the expected risk is equivalent to the minimization of the partial loss which can be evaluated on partially annotated examples. We proposed an instance of the partial learning for the class of linear structured output classifiers which we call Partial-SO-SVM. The Partial-SO-SVM algorithm leads to a hard non-convex optimization problem. We provide an algorithm solving the Partial-SO-SVM problem approximately using an additional prior knowledge about the problem. We demonstrated effectiveness of the proposed method on two real life computer vision problems, namely, the face landmark detection and the image segmentation.



(a) Training image no. 1 and its annotation



(b) Test image no. 1 and predicted segmentation



(c) Test image no. 2 and predicted segmentation

Figure 2: The figure (a) shows an example of a training image and its annotations of varying precision. The precision of the annotations is reciprocal to the width of the unlabeled band (denoted by black) between segments whose width varies from $r = 0$ (pixel precise) to $r = 30$. The figures (b) and (c) show examples of the test images and their segmentation estimated by the MN classifiers trained from examples with annotation of a varying precision.

annotation	$r = 0$	$r = 5$	$r = 10$	$r = 15$	$r = 20$	$r = 25$	$r = 30$
training error [%]	1.52 ± 0.36	1.70 ± 0.29	1.91 ± 0.26	2.12 ± 0.25	2.21 ± 0.23	2.23 ± 0.16	2.36 ± 0.17
testing error [%]	4.21 ± 0.57	4.37 ± 0.34	4.49 ± 0.39	4.61 ± 0.29	4.63 ± 0.34	4.46 ± 0.35	4.52 ± 0.37

Table 2: The table shows results of the SM-PMNF classifier on the cow images from the MSCR database. The first row identifies the used annotation with varying precision from $r = 0$ (pixel precise) to $r = 30$ (the width of the unlabeled band between the segments was 30 pixels). The second and the third line show the training and testing error meassured in the percentage of missclassified pixels.

There are two directions we want for follow in the future. First, the proposed formulation of the partial learning has several limiting assumptions, namely, the output label set must be Cartesian product of label sets of individual objects and the loss function must be additively decomposable over the objects. The goal is to provide more generic formulation which would not rely on the two assumptions. Second, an optimization algorithm solving the Partial-SO-SVM problem with a certificate of optimality remains an open problem that needs to be addressed in the future.

References

[1] Yuille A.L. and Rangarajan A. The concave-convex procedure. *Neural Computation*, 15(4):915–936, 2003.

[2] Antoniuk K., Franc V., and Hláváč V. Learning markov networks by analytic center cutting plane method. In *The International Conference on Pattern Recognition (ICPR)*, 2012. ACCEPTED.

[3] Cour T., Sapp B., and Taskar B. Learning from partial labels. *Journal of Machine Learning Research*, 12:1501–1536, 2011.

[4] I. Tsochantaridis, T. Joachims, T. Hofmann, Y. Altun, and Y. Singer. Large margin methods for structured and interdependent output variables. *Journal of Machine Learning Research*, 6:1453–1484, 2005.

[5] Michal Uřičář, Vojtěch Franc, and Václav Hlaváč. Detector of facial landmarks learned by the structured output SVM. In Gabriela Csurka and José Braz, editors, *VISAPP '12: Proceedings of the 7th International Conference on Computer Vision Theory and Applications*, volume 1, pages 547–556, Portugal, February 2012. SciTePress — Science and Technology Publications.

[6] Lou X. and Hamprecht F.A. Structured learning from partial annotations. In *The International Conference on Machine Learning*, 2012.