Bilevel Optimization with Nonsmooth Lower Level Problems

Peter Ochs[1], René Ranftl[2], Thomas Brox[1], Thomas Pock[2,3]

[1] Computer Vision Group, University of Freiburg, Germany
{ochs,brox}@cs.uni-freiburg.de
[2] Institute for Computer Graphics and Vision, Graz University of Technology, Austria
[3] Digital Safety & Security Department, AIT Austrian Institute of Technology GmbH, 1220 Vienna, Austria
{ranftl,pock}@icg.tugraz.at

Abstract. We consider a bilevel optimization approach for parameter learning in nonsmooth variational models. Existing approaches solve this problem by applying implicit differentiation to a sufficiently smooth approximation of the nondifferentiable lower level problem. We propose an alternative method based on differentiating the iterations of a nonlinear primal–dual algorithm. Our method computes exact (sub)gradients and can be applied also in the nonsmooth setting. We show preliminary results for the case of multi-label image segmentation.

1 Introduction

Many problems in imaging applications and computer vision are approached by variational methods. The solutions are modeled as a state of minimal energy of a function(al). Deviations from multiple model assumptions are penalized by a higher energy. This immediately comes with an important question, namely, the relative importance of the individual assumptions. As it is traditionally hard to manually select the weights, we consider an automatic approach cast as a bilevel optimization problem—an optimization problem that consists of an upper and a lower level. The upper level tries to minimize a certain loss function with respect to the sought set of hyper-parameters. The quantification of the quality of a set of hyper-parameters is only given via the output of the lower level problem. The lower level problem models a specific computer vision task, given a set of hyper-parameters.

Present optimization algorithms for bilevel learning require the lower level problem to be twice differentiable. This limits the flexibility of the approach. For example, in computer vision only a smoothed version of the total variation can be used, whereby favorable properties are lost. Figure 1 plots the energy of a bilevel learning problem and shows the effect of smoothing the lower problem.

In some sense the requirement of regularized models in the lower level problem is a step back in time. In the last decades, people have put a lot of effort to efficiently solve also nonsmooth problems. The main driving force was that nonsmooth energies provide better solutions for many practical problems,

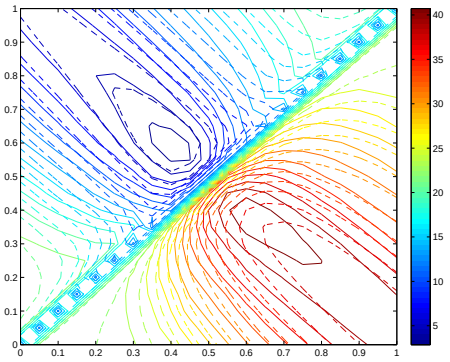

Fig. 1. Contour plot of the energy of a bilevel learning problem with two parameters. The dashed contours correspond to the same learning problem as the solid contours but with a smoothed lower level energy. Usually, gradient descent like schemes are used to find the optimal parameters. We propose a way to compute gradient directions directly on the original problem (solid lines), instead of the smoothed problem (dashed lines) where gradient directions can be completely wrong.

Why not to make use of these powerful optimization tools for bilevel learning? In this paper, we fill the gap between variational bilevel learning and the use of nonsmooth variational models in the lower level problem. The applicability of the developed technique is shown exemplarily for multi-label segmentation, which poses a difficult nonsmooth optimization problem.

2 Related Work

We consider a bilevel optimization problem for parameter learning of the form as considered in [1, 2]. This model for parameter learning is motivated from [3, 4]. The authors argue that the bilevel optimization approach has several advantages compared to classical probabilistic learning methods. In fact, their approach circumvents the problem of computing the partition function of the probability distribution, which is usually not tractable. Earlier, influential approaches are the tree-based bounds of Wainwright et al. [5], Hinton's contrastive divergence method [6] and discriminative learning of graphical models [7, 8].

A generic approach for hyper-parameter optimization is to sample the upper level loss function and regress its shape using Gaussian processes [9] or Random Forests [10]. Since optimization is not based on gradients, it does not require any smoothness of the lower level problem. It rather makes assumptions about the shape of the loss function. This approach is currently limited to the optimization of a moderate number of parameters. Sampling the loss function becomes increasingly demanding if a large number of parameters have to be optimized. Eggensperger [11], for example, reports problem sizes of a few hundred parameters which can be tackled using the generic approach, whereas the bilevel

approach that we consider in this work was successfully applied to problems with up to 30000 parameters [12].

Bilevel optimization was considered for task specific sparse analysis prior learning [13] and applied to signal restoration. In [14, 15] a bilevel approach was used to learn a model of natural image statistics, which was then applied to various image restoration tasks. Recently, it was used for the end-to-end training of a Convolutional Neural Network (CNN) and a graphical model for binary image segmentation [12].

So far all bilevel approaches required the lower level problem to be differentiable; Nonsmooth problems have to be handled using smooth approximations. In [3, 4] differentiability is used in combination with implicit differentiation to analytically differentiate the (upper level) loss function with respect to the parameters. In [1] an efficient semi-smooth Newton method is proposed. In contrast to these approaches the method that we propose can solve bilevel learning problems with a nonsmooth lower level problem.

The procedure of our method is similar to that in [16]. The idea is to directly differentiate the update step of an algorithm that solves the lower level problem with respect to the parameters. Domke [17] applied algorithmic differentiation to derive gradients of truncated gradient based optimization schemes. In contrast to our method, this approach requires to store every intermediate result of the optimization algorithm, which results in a huge memory demand. In [16] the lower level problem is approximated with quadratic majorizers and thus is differentiable by construction. A similar approach was proposed earlier in [18].

Recently, the primal–dual (PD) algorithm from Chambolle and Pock [19] was extended to incorporate Bregman proximity functions [20]. The Bregman proximity function is key in this paper. It allows us to solve a nonsmooth lower level problem with a PD algorithm having differentiable update rules. In [21], in the setting of unbiased risk estimation and parameter selection, iterative (weak) differentiation of Euclidean proximal splitting algorithms is studied.

3 The Bilevel Learning Problem

The bilevel learning problem considered in this paper is the following:

$$\begin{aligned} &\min_{\vartheta} \ \mathcal{L}(x(\vartheta)) \\ &s.t.\ x(\vartheta) \in \arg\min_{x \in \mathbb{R}^N} E(x, \vartheta) \end{aligned} \tag{1}$$

The continuously differentiable function $\mathcal{L}\colon \mathbb{R}^N \to \mathbb{R}_+$ is a loss function describing the discrepancy between a solution $x^*(\vartheta) \in \mathbb{R}^N$ of the lower level problem for a specific set of parameters $\vartheta \in \mathbb{R}^P$ and the training data. The goal is to learn optimal parameters for the lower level problem, given by the proper lower semi-continuous energy $E\colon \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}_+$.

If the lower problem can be explicitly solved for $x^*(\vartheta)$, then the bilevel problem reduces to a single level problem. However, this construction is not always

possible. In that case, implicit differentiation can be used to find a descent direction of $\mathcal{L}(x(\vartheta))$ with respect to ϑ. This is essential for a gradient based optimization method, like it is used in [3], however, twice continuous differentiability of the lower problem is required. We briefly recap the well-known idea before we propose a way to waive this requirement.

3.1 Bilevel Optimization via Implicit Differentiation

The optimality condition of the lower level problem is $\frac{\partial}{\partial x} E(x, \vartheta) = 0$, which under some conditions implicitly defines a function $x^*(\vartheta)$. Let us define $F(x, \vartheta) = \frac{\partial}{\partial x} E(x, \vartheta)$. As we assume that the problem $\min_x E(x, \vartheta)$ has a solution, there is (x^*, ϑ') such that $F(x^*, \vartheta') = 0$. Then the implicit function theorem says that, if F is continuously differentiable and the matrix $\frac{\partial}{\partial x} F(x^*, \vartheta')$ is invertible, there exists an explicit function $X \colon \vartheta \mapsto x(\vartheta)$ in a neighborhood of (x^*, ϑ'). Moreover, the function X is continuously differentiable and

$$\frac{\partial X}{\partial \vartheta}(\vartheta) = \left(-\frac{\partial F}{\partial x}(X(\vartheta), \vartheta) \right)^{-1} \frac{\partial F}{\partial \vartheta}(X(\vartheta), \vartheta) \,.$$

Back-substituting $F = \frac{\partial}{\partial x} E$ and using the Hessian $H_E(X(\vartheta), \vartheta) = \frac{\partial^2 E}{\partial x^2}$ yields

$$\frac{\partial X}{\partial \vartheta}(\vartheta) = -(H_E(X(\vartheta), \vartheta))^{-1} \frac{\partial^2 E}{\partial \vartheta \partial x}(X(\vartheta), \vartheta) \,. \tag{2}$$

The requirement for using (2) from the implicit function theorem is the continuous differentiability of $\frac{\partial}{\partial x} E$ and the invertibility of H_E. Applying the chain rule for differentiation the derivative of the loss function $\mathcal{L}$ of (1) w.r.t. ϑ is

$$\frac{\partial}{\partial \vartheta} \mathcal{L}(x(\vartheta)) = -\frac{\partial \mathcal{L}}{\partial x}(x(\vartheta)) \Big(H_E(X(\vartheta), \vartheta) \Big)^{-1} \frac{\partial^2 E}{\partial \vartheta \partial x}(X(\vartheta), \vartheta) \,. \tag{3}$$

A clever way of setting parentheses avoids explicit inversion of the Hessian matrix [22]. For large problems iterative solvers are required, however.

4 Bilevel Optimization with Nonsmooth Functions

In this section, we resolve the requirement of twice continuous differentiability of the lower level problem. The coarse idea is quite simple: even if the lower level problem is nondifferentiable, there can be algorithms with a differentiable update rule. Let $\mathcal{A}$ and $\mathcal{A}^{(n)} \colon \mathbb{R}^N \times \mathbb{R}^P \to \mathbb{R}^N$ describe one or n iterations, respectively, of algorithm $\mathcal{A}$ for minimizing E in (1). For a fixed $n \in \mathbb{N}$, we replace (1) by

$$\begin{aligned} &\min_{\vartheta} \ \mathcal{L}(x(\vartheta)) \\ &s.t. \ x(\vartheta) = \mathcal{A}^{(n)}(x^0, \vartheta) \,, \end{aligned} \tag{4}$$

where x^0 is some initialization of the algorithm. As the algorithm $\mathcal{A}$ is chosen to solve the (original) lower level problem in (1), we expect it to yield, for each ϑ, a solution $x^{(n)}(\vartheta) \to x^*(\vartheta)$ with $E(\mathcal{A}^{(n)}(x^0, \vartheta), \vartheta) \to \min_x E(x, \vartheta)$ for $n \to \infty$.

An interesting aspect of this approach is that, for a fixed n, the differentiation of $\mathcal{L}$ w.r.t. ϑ is exact; No additional approximation is required. In this way, the algorithm for solving the lower level problem learns parameters that yield an optimal solution after exactly n iterations.

Depending on the problem structure of $\min_x E(x, \vartheta)$ different algorithms can be chosen. We use the flexible PD algorithm from [20], which extends [19] to proximal terms involving Bregman distances. Using this technique, iterations can be made differentiable without requiring differentiability of the energy.

4.1 A Primal–Dual Algorithm with Bregman Distances

We consider the convex–concave saddle-point problem

$$\min_x \max_y \langle Kx, y\rangle + f(x) + g(x) - h^*(y)\,,$$

which is derived from $\min_x f(x) + g(x) + h(Kx)$. One iteration of the PD algorithm [20] reads $(\hat{x}, \hat{y}) = \mathcal{PD}_{\tau,\sigma}(\bar{x}, \bar{y}, \tilde{x}, \tilde{y})$ or

$$\begin{aligned} \hat{x} &= \mathcal{PD}^x_\tau := \arg\min_x f(\bar{x}) + \langle \nabla f(\bar{x}), x - \bar{x}\rangle + g(x) + \langle Kx, \tilde{y}\rangle + \tfrac{1}{\tau} D_x(x, \bar{x}) \\ \hat{y} &= \mathcal{PD}^y_\sigma := \arg\min_y h^*(y) - \langle K\tilde{x}, y\rangle + \tfrac{1}{\sigma} D_y(y, \bar{y})\,, \end{aligned} \tag{5}$$

where $\mathcal{PD}^x_\tau = \mathcal{PD}^x_\tau(\bar{x}, \bar{y}, \tilde{x}, \tilde{y})$ (the same for $\mathcal{PD}^y_\sigma$) with step size parameter σ and τ. The step size parameter must be chosen according to $(\tau^{-1} - L_f)\sigma^{-1} \geq L^2$ where $L = \|K\|$ is the operator norm of K and L_f is the Lipschitz constant of ∇f. The Bregman function $D_x(x, \bar{x}) = \psi_x(x) - \psi_x(\bar{x}) - \langle \nabla \psi_x(\bar{x}), x - \bar{x}\rangle$ is generated by a 1-convex function ψ_x satisfying the requirements and properties in [20] (the same for D_y).

4.2 Primal–Dual Algorithm for Bilevel Learning

Although we assume $\mathcal{A} := \mathcal{PD}_{\tau,\sigma}$ to be differentiable, we do not require it for the lower energy in (4). This allows us to differentiate $\mathcal{A}$ with respect to the parameters. Using the chain rule iterations can be processed successively. A single PD step reads $\frac{\partial}{\partial\vartheta}(\hat{x}(\vartheta), \hat{y}(\vartheta)) = \frac{\partial}{\partial\vartheta}\mathcal{PD}_{\tau,\sigma}(\bar{x}(\vartheta), \bar{y}(\vartheta), \tilde{x}(\vartheta), \tilde{y}(\vartheta))$ where

$$\frac{\partial \mathcal{PD}^x_\tau}{\partial \vartheta} = \frac{\partial \mathcal{PD}^x_\tau}{\partial \bar{x}}\frac{\partial \bar{x}}{\partial \vartheta}(\vartheta) + \frac{\partial \mathcal{PD}^x_\tau}{\partial \bar{y}}\frac{\partial \bar{y}}{\partial \vartheta}(\vartheta) + \frac{\partial \mathcal{PD}^x_\tau}{\partial \tilde{x}}\frac{\partial \tilde{x}}{\partial \vartheta}(\vartheta) + \frac{\partial \mathcal{PD}^x_\tau}{\partial \tilde{y}}\frac{\partial \tilde{y}}{\partial \vartheta}(\vartheta)\,, \tag{6}$$

and we dropped the dependency of $\mathcal{PD}^x_\tau$ on $(\bar{x}(\vartheta), \bar{y}(\vartheta), \tilde{x}(\vartheta), \tilde{y}(\vartheta))$ for clarity. The analogous expression holds for $\mathcal{PD}^y_\sigma$. As the functions $\bar{x}(\vartheta)$, $\bar{y}(\vartheta)$, $\tilde{x}(\vartheta)$ and $\tilde{y}(\vartheta)$ are simple combinations (products with scalars and sums) of the output of the previous PD iteration, the generalization to n iterations is straightforward.

5 Application to Multi-Label Segmentation

In this section, we show how the developed abstract idea is applied in practice. Before the actual bilevel learning problem is presented, we introduce the multi-label segmentation model. Then, the standard (nondifferentiable) PD approach to this problem, our (differentiable) formulation, and the PD algorithm for the smoothed energy (required by the implicit differentiation framework) are shown.

5.1 Model and Discretization

Given a cost tensor $\mathfrak{c} \in X^{N_l}$, where $X = \mathbb{R}^{N_x N_y}$, that assigns to each pixel (i,j) and each label k, $i = 1,\dots,N_x$, $j = 1,\dots,N_y$, $k = 1,\dots,N_l$, a cost $\mathfrak{c}^k_{i,j}$ for the pixel taking label k. We often identify $\mathbb{R}^{N_x \times N_y}$ with $\mathbb{R}^{N_x N_y}$ by $(i,j) \mapsto i+(j-1)N_x$ to simplify the notation. The sought segmentation $u \in X^{N_l}_{[0,1]}$, where $X_{[0,1]} = [0,1]^{N_x N_y} \subset X$, is represented by a binary vector for each label. As a regularizer for a segment's plausibility we measure the boundary length using the total variation (TV). The discrete derivative operator $\nabla\colon X \to Y$, where we use the shorthand $Y := X \times X$ (elements from Y are considered as column vectors), is defined as (let the pixel dimension be 1×1):

$$(\nabla u^k)_{i,j} := \begin{pmatrix} (\nabla u^k)^x_{i,j} \\ (\nabla u^k)^y_{i,j} \end{pmatrix} \in Y (= \mathbb{R}^{2N_x N_y}), \quad Du := (\nabla u^1, \dots, \nabla u^{N_l})$$

$$(\nabla u^k)^x_{i,j} := \begin{cases} u^k_{i+1,j} - u^k_{i,j}, & \text{if } 1 \le i < N_x, 1 \le j \le N_y \\ 0, & \text{if } i = N_x, 1 \le j \le N_y \end{cases}$$

$(\nabla u^k)^y_{i,j}$ is defined analogously. From now on, we work with the image as a vector indexed by $\mathbf{i} = 1,\dots,N_x N_y$. Let elements in Y be indexed with $\mathbf{j} = 1,\dots,2N_x N_y$. Let the inner product in X and Y be given, for $u^k, v^k \in X$ and $p^k, q^k \in Y$, as: $\langle u^k, v^k \rangle_X := \sum_{\mathbf{i}=1}^{N_x N_y} u^k_{\mathbf{i}} v^k_{\mathbf{i}}$ and $\langle p^k, q^k \rangle_Y := \sum_{\mathbf{j}=1}^{2N_x N_y} p^k_{\mathbf{j}} q^k_{\mathbf{j}}$, $\langle u, v \rangle_{X^{N_l}} := \sum_{k=1}^{N_l} \langle u^k, v^k \rangle_X$ and $\langle p, q \rangle_{Y^{N_l}} := \sum_{k=1}^{N_l} \langle p^k, q^k \rangle_Y$. The (discrete, anisotropic) TV norm is given by $\|Du\|_1 := \sum_{k=1}^{N_l} \sum_{\mathbf{j}=1}^{2N_x N_y} |(\nabla u^k)_{\mathbf{j}}|$, where $|\cdot|$ is the absolute value. In the following, the iteration variables $\mathbf{i} = 1,\dots,N_x N_y$ and $\mathbf{j} = 1,\dots,2N_x N_y$ always run over these index sets, thus we drop the specification; the same for $k = 1,\dots,N_l$. We define the pixel-wise nonnegative unit simplex

$$\Delta^{N_l} := \{u \in X^{N_l} \,|\, \forall(\mathbf{i},k)\colon 0 \le u^k_{\mathbf{i}} \le 1 \text{ and } \forall \mathbf{i}\colon \textstyle\sum_k u^k_{\mathbf{i}} = 1\}, \tag{7}$$

and the pixel-wise (closed) ℓ_∞-unit ball around the origin

$$B^{\ell_\infty}_1(0) := \{p \in Y^{N_l} \,|\, \forall(\mathbf{j},k)\colon |p^k_{\mathbf{j}}| \le 1\}.$$

Finally, the segmentation model reads

$$\min_{u \in X^{N_l}} \ \langle \mathfrak{c}, u \rangle_{X^{N_l}} + \|Du\|_1, \quad s.t.\ u \in \Delta^{N_l}. \tag{8}$$

This model and the following reformulation as a saddle-point problem are well known (see e.g. [19])

$$\min_{u\in X^{N_l}} \max_{p\in Y^{N_l}} \langle Du,p\rangle_{Y^{N_l}} + \langle u,\mathfrak{c}\rangle_{X^{N_l}} \;, \quad s.t.\; u\in \Delta^{N_l},\; p\in B_1^{\ell^\infty}(0)\,. \tag{9}$$

5.2 Parameter Learning Setting

We consider (8) where the cost is given for each label k by $\mathfrak{c}_{\mathbf{i}}^k = \lambda(\mathfrak{I}_{\mathbf{i}} - \vartheta^k)^2$, where $\mathfrak{I}\in X$ is the image to be segmented and λ is a positive balancing parameter. ϑ^k can be interpreted as the mean value of the region with label k.

The training set consists of N_T images $\mathfrak{I}^1,\dots,\mathfrak{I}^{N_T}\in X$ and corresponding ground truth segmentations $\mathfrak{g}^1,\dots,\mathfrak{g}^{N_T}$. The ground truths are generated by solving (8) with $(\mathfrak{c}^t)_{\mathbf{i}}^k = \lambda(\mathfrak{I}_{\mathbf{i}}^t - \hat{\vartheta}^k)^2$ for each $t\in\{1,\dots,N_T\}$ and predefined parameters $\hat{\vartheta}^1,\dots,\hat{\vartheta}^{N_l}$.

We consider an instance of the general bilevel optimization problem (1):

$$\begin{aligned} &\min_{\vartheta\in\mathbb{R}^{N_l}} \frac{1}{2}\sum_{t=1}^{N_T} \|u(\vartheta,\mathfrak{I}^t) - \mathfrak{g}^t\|_2^2 \\ &s.t.\; u(\vartheta,\mathfrak{I}^t) = \arg\min_{u\in X^{N_l}} E(u,\mathfrak{c}^t)\,, \qquad (\mathfrak{c}^t)_{\mathbf{i}}^k = \lambda(\mathfrak{I}_{\mathbf{i}}^t - \vartheta^k)^2\,. \end{aligned} \tag{10}$$

The goal is to learn the parameters (the mean values) ϑ^k and try to recover $\hat{\vartheta}^k$. The energy E in the lower level problem is (8).

5.3 The Standard Primal–Dual Algorithm

Problem (8) can be solved using the PD algorithm from (5). The standard way to apply it is by setting $x=u$, $y=p$, $f\equiv 0$, $g(u) = \langle u,\mathfrak{c}\rangle_{X^{N_l}} + \delta_{\Delta^{N_l}}(u)$, and $h^*(p) = \sum_k\sum_{\mathbf{j}} \delta_{[-1,1]}(p_{\mathbf{j}}^k)$, where δ_C is the indicator function of the convex set C. Furthermore, the Bregman functions are the squared Euclidean distance (for primal and dual update) and the constraints of the primal variable are incorporated in the proximal step. It reads

$$\begin{aligned} \hat{u} &= \mathit{\Pi}_{\Delta^{N_l}}\left(\bar{u} - \tau D^\top\tilde{p} - \tau\mathfrak{c}\right) \\ \hat{p} &= \mathit{\Pi}_{B_1^{\ell^\infty}(0)}\left(\bar{p} + \sigma D u\right)\,, \end{aligned} \tag{11}$$

where $\mathit{\Pi}_C$ denotes the orthogonal projection operator onto the set C. As these projections are nonsmooth functions, they are not suited for our framework.

5.4 A Primal–Dual Algorithm with Bregman Proximity Function

A differentiable PD iteration can be derived using the Bregman function

$$D_x(u,\bar{u}) = \frac{1}{2}\sum_k\sum_{\mathbf{i}} u_{\mathbf{i}}^k(\log(u_{\mathbf{i}}^k) - \log(\bar{u}_{\mathbf{i}}^k)) - u_{\mathbf{i}}^k + \bar{u}_{\mathbf{i}}^k\,,$$

which is generated by $\psi_x(u) = \frac{1}{2}\sum_{k,\mathbf{i}} u_{\mathbf{i}}^k \log(u_{\mathbf{i}}^k)$. The key idea for choosing this Bregman function is that it takes finite values only for nonnegative coordinates. As a consequence the nonnegativity constraint in the primal update step can be dropped and the projection is given by a simple analytic expression:

$$\forall(k,\mathbf{i}):\quad \hat{u}_{\mathbf{i}}^k = \frac{\exp(-2\tau(\nabla^\top \tilde{p}^k)_{\mathbf{i}} - 2\tau \mathfrak{c}_{\mathbf{i}}^k)\bar{u}_{\mathbf{i}}^k}{\sum_{k'=1}^{N_l} \exp(-2\tau(\nabla^\top \tilde{p}^{k'})_{\mathbf{i}} - 2\tau \mathfrak{c}_{\mathbf{i}}^{k'})\bar{u}_{\mathbf{i}}^{k'}}\,. \tag{12}$$

For the dual update step we use the Bregman proximity function

$$\begin{aligned} D_y(p,\bar{p}) = \frac{1}{2}\sum\nolimits_k \sum\nolimits_{\mathbf{j}} \; & (1-p_{\mathbf{j}}^k)(\log(1-p_{\mathbf{j}}^k) - \log(1-\bar{p}_{\mathbf{j}}^k)) - p_{\mathbf{j}}^k + \bar{p}_{\mathbf{j}}^k \\ & + (1+p_{\mathbf{j}}^k)(\log(1+p_{\mathbf{j}}^k) - \log(1+\bar{p}_{\mathbf{j}}^k)) - p_{\mathbf{j}}^k + \bar{p}_{\mathbf{j}}^k\,, \end{aligned}$$

which is generated by $\psi_y(p) = \frac{1}{2}\sum_k \sum_{\mathbf{j}} (1+p_{\mathbf{j}}^k)\log(1+p_{\mathbf{j}}^k) + (1-p_{\mathbf{j}}^k)\log(1-p_{\mathbf{j}}^k)$. It takes finite values only within the feasible set $[-1,1]$ for each coordinate.

$$\forall(k,\mathbf{j}):\quad \hat{p}_{\mathbf{j}}^k = \frac{\exp(2\sigma(\nabla \tilde{u}^k)_{\mathbf{j}}) - \frac{1-\bar{p}_{\mathbf{j}}^k}{1+\bar{p}_{\mathbf{j}}^k}}{\exp(2\sigma(\nabla \tilde{u}^k)_{\mathbf{j}}) + \frac{1-\bar{p}_{\mathbf{j}}^k}{1+\bar{p}_{\mathbf{j}}^k}} \tag{13}$$

emerges as the resulting update step. (12) and (13) define the update function $(\hat{u},\hat{p}) = \mathcal{PD}_{\tau,\sigma}(\bar{u},\bar{p},\tilde{u},\tilde{p})$ for the PD algorithm, which is differentiable.

5.5 A Smoothed Parameter Learning Problem

The method of implicit differentiation requires the lower level problem of (10) to be twice differentiable. As in [12] for binary segmentation, the domain constraint $u_{\mathbf{i}}^k \in [0,1]$ is incorporated via a log barrier $\mu\sum_{k,\mathbf{i}}(\log(u_{\mathbf{i}}^k) + \log(1-u_{\mathbf{i}}^k))$ with $\mu < 0$ and instead of the TV for each label function the smooth Charbonnier function $\|Du\|_\varepsilon := \sum_k \sum_{\mathbf{j}} ((\nabla u^k)_{\mathbf{j}}^2 + \varepsilon^2)^{\frac{1}{2}}$ with $\varepsilon > 0$ is used. The simplex constraint (7) is incorporated using a Lagrange multiplier $\rho \in X$, such that the smoothed Lagrangian reads

$$E_\varepsilon(u,\rho) := \langle \mathfrak{c}, u\rangle_{X^{N_l}} + \|Du\|_\varepsilon + \Big\langle \rho, \sum\nolimits_k u^k - \mathbf{1}\Big\rangle_X + \mu\sum\nolimits_{k,\mathbf{i}}(\log(u_{\mathbf{i}}^k) + \log(1-u_{\mathbf{i}}^k))\,,$$

where $(1,\dots,1)^\top =: \mathbf{1} \in X$. As the Hessian matrix of E_ε with respect to (u,ρ) needs to be computed at the optimum of $\min_u \max_\rho E_\varepsilon(u,\rho)$, we seek for its efficient optimization. We use the PD algorithm [20] (see (5)) with Euclidean proximity functions by setting $f(u) = \|Du\|_\varepsilon$, $g(u) = \langle \mathfrak{c}, u\rangle_{X^{N_l}} + \mu\sum_{k,\mathbf{i}}(\log(u_{\mathbf{i}}^k) + \log(1-u_{\mathbf{i}}^k))$, $h^*(\rho) = \langle \rho, \mathbf{1}\rangle_X$, and K such that $Ku := \sum_k u^k$. The Lipschitz constant of ∇f is $L_f = 8/\varepsilon$, the operator norm is $L = \|K\| = N_l$, and the strong convexity modulus of g is -8μ. These properties allow us to use the accelerated PD algorithm. Sadly, the proximal map of g requires to solve (coordinate-wise) for the unique root of a cubic polynomial in $[0,1]$, which is expensive.

Discussion of the smoothed model. Opposed to our approach, smoothing the energy has several disadvantages: (1) It is only an approximation to the actual energy; (2) additional terms for dealing with constraints are required; (3) the extra variable ρ increases the size of the Hessian matrix of E_ε by $N_x N_y$ to $N_x N_y (N_l + 1)$; (4) the proximal map is costly to solve; and (5) the Lipschitz constant, hence the step size, is directly affected by ε, i.e. by the approximation quality. (5) can be resolved by another approximation. If we set $f = 0$ and dualize the Charbonnier function, the step size becomes independent of ε. However, the proximal map for the Charbonnier function—the same holds for its dual function—is not simple, a numerical solver is required for its minimization.

5.6 Experiment for Parameter Learning

We consider the bilevel optimization problem in (10) with ground truth parameters $(\hat{\vartheta}^1, \hat{\vartheta}^2) = (0.4, 0.6)$. The balancing parameter was set to $\lambda = 20$. The dataset consists of 50 images from the Weizmann horse dataset [23]. Each image was converted to gray scale and downsampled by factor 10. For each image, we generated a segmentation by running 2000 iterations of (11) with the ground truth mean value parameters. Note that this is a numerical toy problem, where we are interested in retrieving the parameters that lead to these segmentations. We are *not* interested in segmentations that correspond to horses.

Figure 1 shows the upper level energy (solid lines) obtained using segmentations for parameters $(\vartheta^1, \vartheta^2)$ sampled on a regular grid. The dashed lines correspond to the smoothed lower level problem with $\varepsilon = 0.1$, $\mu = 10^{-4}$. The energies differ a lot, although this is a simple problem. Reducing ε yields better approximations but also makes the lower level problem harder to solve.

We solve the learning problem with a simple gradient descent method with backtracking initialized at $(0.13, 0.56)$ with a maximum of 50 iterations. Figure 2 compares the convergence of our method with the implicit differentiation approach (implDiff) for different numbers of inner iterations. Our approach reaches the optimum already for 200 iterations. It clearly requires fewer inner iterations than the implDiff method. The segmentations are shown in Figure 3.

As Figure 2 shows, the gradient directions computed with our framework align with the geometric gradient—this is the reason for optimizing with gradient descent—, which is orthogonal to the level lines. The gradients computed with the implDiff framework often point to a different direction. For a small number of inner iterations, the energy computed with the smoothed segmentation model deviates even more from the original energy than in Figure 1. Inverting the poorly conditioned Hessian matrix (by solving a system of equations) amplifies inaccuracies of the lower level solution significantly.

As the original and the smoothed energies have similar minimizers in this two-dimensional example, also the implDiff framework approaches the optimum with more inner iterations. Due to inappropriate step sizes determined by the simple backtracking that we use, our method fails to find the optimum when using 800 inner iterations. With iPiano [24] we found the exact optimum; see also Figure 3. Another option to iPiano is L-BFGS [25].

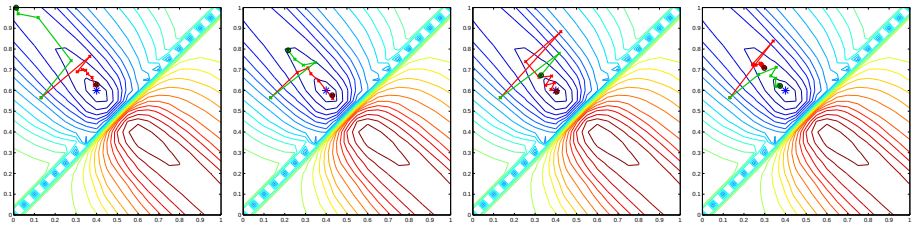

Fig. 2. Convergence of our approach (red line) vs. the implDiff approach (green line) visualized on a contour plot of the two-parameter problem. From left to right: The learning problem is solved with 20, 100, 400, and 800 inner iteration. The gradients computed with our method are orthogonal to the level lines even for few inner iterations.

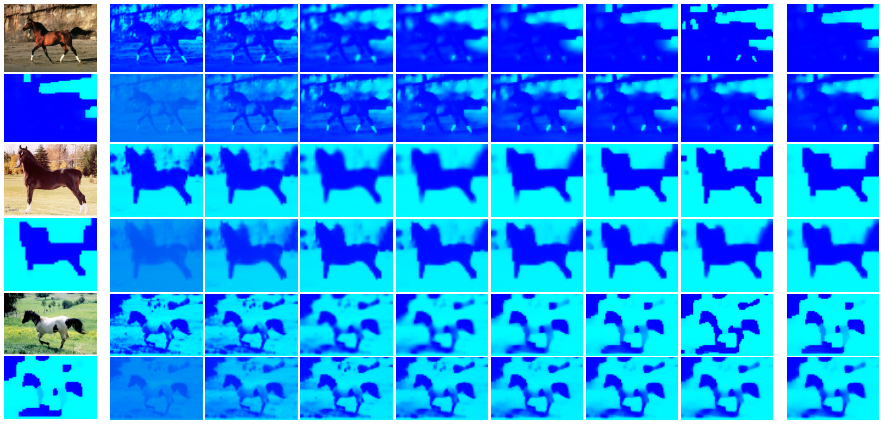

Fig. 3. Row-wise alternating: left column: input sample, ground truth segmentation; right block: our method, implDiff; and from left to right: numbers of inner iterations: 5, 20, 50, 100, 200, 400, 800, and 800 (iPiano) for the two-parameter problem.

Since the parameter learning problem is nonconvex, initialization matters. The initialization that we used was selected among 3 randomly generated proposals, to show a good performance of both approaches. In general our gradient based optimization could be a good complement to zero-order search methods. This will be subject to future work.

We simulate such a scenario by initializing the following 4-label segmentation experiment close to the optimum. We perturb the ground truth parameters $(0.17, 0.37, 0.42, 0.98)$ randomly with numbers drawn uniformly in $[-0.1, 0.1]$. λ is set to 120, and 400 inner iterations are performed on the single training example in Figure 4. The final Euclidean distance, the error, between our solution parameters and the ground truth parameters is about $0.4 \cdot 10^{-2}$, and for implDiff it is $4.75 \cdot 10^{-2}$. Corresponding segmentations are shown in Figure 4.

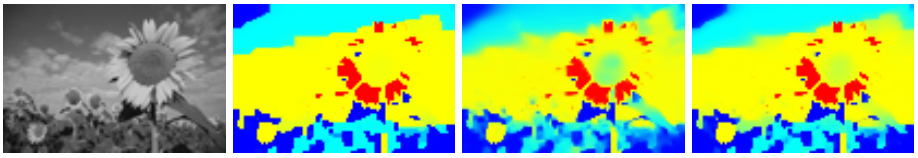

Fig. 4. Parameter learning problem and results for sunflowers (102×68). From left to right: input image, ground truth segmentation with mean values $(0.17, 0.37, 0.42, 0.98)$, segmentation obtained with implDiff, and our method, both with 400 inner iterations.

6 Conclusion

We considered a bilevel optimization problem for parameter learning and proposed a way to overcome one of its main drawbacks. Solving the problem with gradient based methods requires to compute the gradient with respect to the parameters and thus also requires (twice) differentiability of the lower level problem. With our approach the lower level problem can be nondifferentiable; Only a differentiable mapping from the parameters to a solution of the lower level problem is needed. We propose to use the iteration mapping of a recently proposed primal–dual algorithm with Bregman proximity functions as such a mapping. Fixing a number of iterations, the computation of gradients w.r.t. the parameters is exact. Our algorithm learns to yield optimal parameters when using exactly this number of iterations. The abstract idea was exemplified on the (nonsmooth) multi-label segmentation problem.

Acknowledgment

Peter Ochs and Thomas Brox acknowledge support by DFG grant BR 3815/8-1 in the SPP 1527 Autonomous Learning. René Ranftl and Thomas Pock acknowledge support from the Austrian science fund under the ANR-FWF project "Efficient algorithms for nonsmooth optimization in imaging", No. I1148 and the FWF-START project "Bilevel optimization for Computer Vision", No. Y729.

References

1. Kunisch, K., Pock, T.: A bilevel optimization approach for parameter learning in variational models. SIAM Journal on Imaging Sciences **6**(2) (2013) 938–983
2. Reyes, J.C.D.L., Schönlieb, C.B.: Image denoising: Learning noise distribution via pde-constrained optimisation. Inverse Problems and Imaging **7** (2013) 1183–1214
3. Samuel, K., Tappen, M.: Learning optimized MAP estimates in continuously-valued MRF models. In: International Conference on Computer Vision and Pattern Recognition (CVPR). (2009) 477–484
4. Tappen, M., Samuel, K., Dean, C., Lyle, D.: The logistic random field–a convenient graphical model for learning parameters for MRF-based labeling. In: International Conference on Computer Vision and Pattern Recognition (CVPR). (2008) 1–8
5. Wainwright, M., Jaakkola, T., Willsky, A.: MAP estimation via agreement on (hyper)trees: Message-passing and linear programming approaches. IEEE Transactions on Information Theory **51** (2002) 3697–3717

6. Hinton, G.: Training products of experts by minimizing contrastive divergence. Neural Computation **14**(8) (2002) 1771–1800
7. Taskar, B., Chatalbashev, V., Koller, D., Guestrin, C.: Learning structured prediction models: a large margin approach. In: International Conference on Machine Learning (ICML). (2005) 896–903
8. LeCun, Y., Huang, F.: Loss functions for discriminative training of energy-based models. In: International Workshop on Artificial Intelligence and Statistics. (2005)
9. Snoek, J., Larochelle, H., Adams, R.P.: Practical Bayesian Optimization of Machine Learning Algorithms. In: Advances in Neural Information Processing Systems (NIPS). (2012) 2951–2959
10. Hutter, F., Hoos, H.H., Leyton-Brown, K.: Sequential model-based optimization for general algorithm configuration. In: Proceedings of the 5th International Conference on Learning and Intelligent Optimization. LION (2011) 507–523
11. Eggensperger, K., Feurer, M., Hutter, F., Bergstra, J., Snoek, J., Hoos, H., Leyton-Brown, K.: Towards an empirical foundation for assessing bayesian optimization of hyperparameters. In: NIPS workshop. (2013)
12. Ranftl, R., Pock, T.: A deep variational model for image segmentation. In: German Conference on Pattern Recognition (GCPR). (2014) 107–118
13. Peyré, G., Fadili, J.: Learning analysis sparsity priors. In: Proceedings of Sampta. (2011)
14. Chen, Y., Pock, T., Ranftl, R., Bischof, H.: Revisiting loss-specific training of filter-based MRFs for image restoration. In: German Conference on Pattern Recognition (GCPR). (2013)
15. Chen, Y., Ranftl, R., Pock, T.: Insights into analysis operator learning: From patch-based sparse models to higher order MRFs. IEEE Transactions on Image Processing **23**(3) (2014) 1060–1072
16. Tappen, M.: Utilizing variational optimization to learn MRFs. In: International Conference on Computer Vision and Pattern Recognition (CVPR). (2007) 1–8
17. Domke, J.: Generic methods for optimization-based modeling. In: International Workshop on Artificial Intelligence and Statistics. (2012) 318–326
18. Geman, D., Reynolds, G.: Constrained restoration and the recovery of discontinuities. IEEE Transactions on Pattern Analysis and Machine Intelligence **14** (1992) 367–383
19. Chambolle, A., Pock, T.: A first-order primal-dual algorithm for convex problems with applications to imaging. Journal of Mathematical Imaging and Vision **40**(1) (2011) 120–145
20. Chambolle, A., Pock, T.: On the ergodic convergence rates of a first-order primal-dual algorithm. Technical report (2014) to appear.
21. Deledalle, C.A., Vaiter, S., Fadili, J., Peyré, G.: Stein Unbiased GrAdient estimator of the Risk (SUGAR) for multiple parameter selection. SIAM Journal on Imaging Sciences **7**(4) (2014) 2448–2487
22. Foo, C.S., Do, C., Ng, A.: Efficient multiple hyperparameter learning for log-linear models. In: Advances in Neural Information Processing Systems (NIPS). Curran Associates, Inc. (2008) 377–384
23. Borenstein, E., Sharon, E., Ullman, S.: Combining top-down and bottom-up segmentation. In: International Conference on Computer Vision and Pattern Recognition Workshop (CVPR). (2004)
24. Ochs, P., Chen, Y., Brox, T., Pock, T.: ipiano: Inertial proximal algorithm for non-convex optimization. SIAM Journal on Imaging Sciences **7**(2) (2014) 1388–1419
25. Liu, D.C., Nocedal, J.: On the limited memory BFGS method for large scale optimization. Mathematical Programming **45**(1) (1989) 503–528