# Learning Adaptive Regularization for Image Labeling Using Geometric Assignment

R. Hühnerbein[1], F. Savarino[1], S. Petra[2], and C. Schnörr[1]

[1] Image and Pattern Analysis Group, Heidelberg University, Germany
[2] Mathematical Imaging Group, Heidelberg University, Germany

**Abstract.** We introduce and study the *inverse problem of model parameter learning for image labeling*, based on the linear assignment flow. This flow parametrizes the assignment of labels to feature data on the assignment manifold through a linear ODE on the tangent space. We show that both common approaches are *equivalent*: either differentiating the continuous system and numerical integration of the state and the adjoint system, or discretizing the problem followed by constrained parameter optimization. Experiments demonstrate how a parameter prediction map based on kernel regression and optimal parameter values, enables the assignment flow to perform *adaptive regularization* that can be directly applied to novel data.

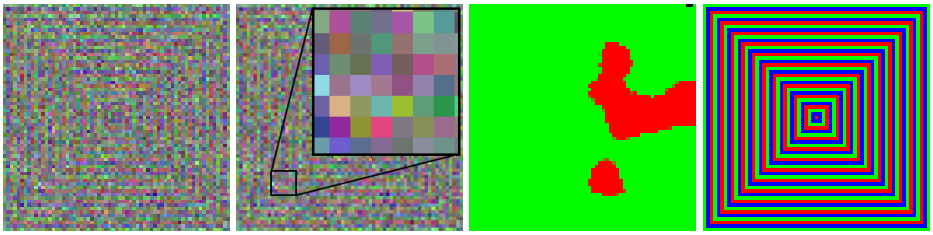
**Keywords:** image labeling, assignment manifold, linear assignment flows, parameter learning, geometric integration, adaptive regularization.

## 1 Introduction

The *image labeling problem*, i.e. the problem to classify images depending on the spatial context, has been thoroughly investigated during the last two decades. While the evaluation (inference) of such models is well understood [10], *learning the parameters* of such models has remained elusive, in particular for models with higher connectivity of the underlying graph. Various sampling-based and other approximation methods exist (cf., e.g. [17] and references therein), but the *relation* between approximations of the *learning problem* on the one hand, and approximations of the subordinate *inference problem* on the other hand, is less well understood [14].

This paper is based on the *assignment flow for image labeling* introduced in [2] and on the *linear* assignment flow introduced and studied in [16]. These flows are induced by dynamical systems that evolve on an elementary statistical manifold, the so-called *assignment manifold*. Regarding the *inference task*, it has been shown recently that the assignment flow can evaluate discrete graphical models [3].

**Contribution.** In this paper, we take a first step towards *learning the model parameters* using the linear assignment flow. These parameters control local geometric averaging as part of the vector field which drives the flow. The problem to *learn* these parameters was raised in [2, Section 5 and Fig. 14]. See Figure 1 below for an illustration. Our contribution can be characterized as follows.



(a) noisy input image (b) close-up view (c) labeling with *uniform* weights (d) labeling with *adaptive* weights

**Fig. 1.** Running the linear assignment flow with *uniform* weights for geometric averaging corresponds – roughly speaking – to labeling with a graphical model that involves a Potts prior (uniform label distances) for regularization. As it is well known, this does not work, e.g., for fine spatial structures contaminated with noise (cf. panels (a), (b) and the result (c)). This paper presents a *general* method for *learning how to regularize image labeling* using linear assignment flows, by Riemannian gradient flows on the manifold of regularization parameters, so as to recognize 'familiar' image structure (panel (d)).

**Exact inference.** The inference problem is solved *exactly* during learning, unlike in work based on discrete graphical models, as discussed above. As a consequence, the 'predictive power' of learned parameters can be expected to be larger.

**Reproducible results.** Our results are based on a simple algorithm for numerical geometric integration and hence are *reproducible*. This contrasts with current research on deep networks and complex software tools for training, that are more powerful in applications but are also less well understood [13].

**Networks as dynamical systems.** Our work ties in with research on networks from a *dynamical systems* point of view, that emanated from [9] in computer science and has also been promoted recently in mathematics [4]. The recent work [7], for example, studied stability issues of discrete-time network dynamics using techniques of numerical ODE integration. The authors adopted the *discretize-then-differentiate* viewpoint on the parameter estimation problem and suggested symplectic numerical integration in order to achieve better stability.

**Commuting modelling diagram.** Our work contrasts in that inference is always *exact* during learning, unlike [7] where learning is based on *approximate* inference. Furthermore, symplectic numerical integration is a *consequence*, in our case, of resolving the dilemma of *what path* to choose in the 'modelling diagram' of Figure 2: the fact that both paths *commute* qualifies our approach as a proper (though rudimentary) method of *optimal control* (cf. [12]).

**Organization of the paper.** Section 2 summarizes the assignment flow, the linear assignment flow and related concepts. Section 3 details our approach: both paths of the diagram of Fig. 2 are worked out and shown to be *equivalent*. Experiments in Section 4 demonstrate that a basic kernel-based regression estimator based on optimal parameter values and corresponding features makes the assignment flow *adaptive* and directly applicable to *novel* data.

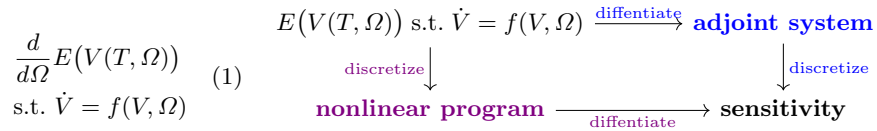



**Fig. 2.** Illustration of the methodological part of this paper. LEFT: The *sensitivity* of an objective function $E$ with respect to parameters $\Omega$ to be estimated, defines a gradient flow on the parameter manifold for adapting initial parameter values. The parameter dependency of $E$ is implicitly given through the linear assignment flow $V(t)$ at the terminal point of time $t = T$, whose state equation depends on $\Omega$. RIGHT: Our approach satisfies the commuting diagram, i.e. *identical* results are obtained either if the continuous problem is differentiated first and than discretized (blue path), or the other way around (violet path).

## 2 Preliminaries

This section summarizes the *assignment flow* and its approximation, the *linear assignment flow*, introduced in [2] and [16], respectively. The latter provides the basis for our appoach to parameter estimation developed in Section 3.

### 2.1 Assignment Flow

Let $G = (I, E)$ be a given undirected graph with vertices $i \in I$ indexing data $\mathcal{F}_I = \{f_i \colon i \in I\} \subset \mathcal{F}$ given in a metric space $(\mathcal{F}, d)$. The edge set $E$ specifies neighborhoods $\mathcal{N}_i = \{k \in I \colon ik = ki \in E\}$ for every pixel $i \in I$ along with positive weight vectors $w_i \in \operatorname{rint} \Delta_{|\mathcal{N}_i|}$, where $\operatorname{rint} \Delta_n = \Delta_n \cap \mathbb{R}^n_{++}$ denotes the relative interior of the probability simplex $\Delta_n$.

Along with $\mathcal{F}_I$, *prototypical data (labels)* $\mathcal{L}_J = \{l_j \in \mathcal{F} \colon j \in J\}$ are given that represent classes $j = 1, \ldots, |J|$. *Supervised image labeling* denotes the task to assign precisely one prototype $l_j$ to each datum $f_i$ in a spatially coherent way. These assignments are represented at each pixel $i$ by probability vectors

$$W_i \in \mathcal{S} := (\operatorname{rint} \Delta_{|J|}, g_{FR}), \quad i \in I \tag{2}$$

on the relative interior of the simplex $\Delta_{|J|}$, that together with the Fisher-Rao metric $g_{FR}$ becomes a Riemannian manifold denoted by $\mathcal{S}$. Collecting all assignment vectors into a strictly positive, row-stochastic matrix

$$W = (W_1, \ldots, W_{|I|})^\top \in \mathcal{W} = \mathcal{S} \times \cdots \times \mathcal{S} \subset \mathbb{R}^{|I| \times |J|} \tag{3}$$

defines a point on the *assignment manifold* $\mathcal{W}$. Image labeling is accomplished by geometrically integrating the *assignment flow* (the r.h.s. is defined below)

$$\dot W = \Pi_W\big(S(W)\big), \qquad W(0) = \mathbb{1}_{\mathcal{W}} := \frac{1}{|J|} \mathbb{1}_{|I|} \mathbb{1}_{|J|}^\top \qquad \text{(barycenter)}, \tag{4}$$

that evolves from the barycenter $W(0)$ towards *pure* assignment vectors, i.e. each vector $W_i$ approaches the $\varepsilon$-neighborhood of some unit vector at some vertex of $\mathcal{S}$ and hence a *labeling* after trivial rounding.

In order to explain the rationale behind (4), we need the following maps based on the affine e-connection of information geometry [1] in place of the Levi-Civita connection on the tangent bundle of the manifolds $\mathcal{S}$ and $\mathcal{W}$: With tangent space $T_0 = T_p\mathcal{S}$ independent of the base point $p \in \mathcal{S}$, we define

$$\mathbb{R}^{|J|} \ni z \mapsto \Pi_p(z) = \big(\operatorname{Diag}(p) - pp^\top\big)z \in T_0, \tag{5a}$$

$$\mathcal{S} \times T_0 \ni (p, v) \mapsto \operatorname{Exp}_p(v) = \frac{e^{\frac{v}{p}}}{\langle p, e^{\frac{v}{p}}\rangle} p \in \mathcal{S}, \tag{5b}$$

$$\mathcal{S} \times \mathcal{S} \ni (p, q) \mapsto \operatorname{Exp}_p^{-1}(q) = \Pi_p \log \frac{q}{p} \in T_0, \tag{5c}$$

$$\mathcal{S} \times \mathbb{R}^{|J|} \ni (p, z) \mapsto \exp_p(z) = \operatorname{Exp}_p \circ \Pi_p(z) = \frac{pe^z}{\langle p, e^z\rangle} \in \mathcal{S}, \tag{5d}$$

where multiplication, subdivision and the exponential function $e^{(\cdot)}$ apply *componentwise* to strictly positive vectors in $\mathcal{S}$. Corresponding maps $\Pi_W, \operatorname{Exp}_W, \exp_W$ in connection with the product manifold (3) are defined analogously, as is the tangent space $\mathcal{T}_0 = T_0 \times \cdots \times T_0$.

The vector field defining the assignment flow on the right-hand side of (4) is defined as follows. Given a metric $d$, data $\mathcal{F}_I$ and labels $\mathcal{L}_J$, distance vectors $D_i = \big(d(f_i, l_1), \ldots, d(f_i, l_{|J|})\big)^\top$ are defined at each pixel $i \in I$ and mapped to the assignment manifold by

$$L(W) = \exp_W(-\tfrac{1}{\rho}D) \in \mathcal{W}, \quad L_i(W_i) = \exp_{W_i}(-\tfrac{1}{\rho}D_i) = \frac{W_i e^{-\frac{1}{\rho}D_i}}{\langle W_i, e^{-\frac{1}{\rho}D_i}\rangle}, \tag{6}$$

where $\rho > 0$ is a user parameter for normalizing the scale of the data. These *likelihood vectors* represent 'data terms' in conventional variational approaches, and they are *spatially regularized* in a way conforming to the geometry of $\mathcal{S}$, to obtain

$$S(W) = \mathcal{G}^\omega(L(W)) \in \mathcal{W}, \quad \mathcal{G}_i^\omega(W) := \operatorname{Exp}_{W_i}\Big(\sum_{k \in \mathcal{N}_i} w_{ik} \operatorname{Exp}_{W_i}^{-1}(W_k)\Big). \tag{7}$$

Note that (7) is *parametrized* by the 'weight patches' $(w_{ik})_{k\in\mathcal{N}_i}$, $i \in I$. **Learning these parameters from data is the subject of this paper.**

The assignment flow (4) now simply *approximates* the *Riemannian gradient ascent flow* $\dot{W} = \nabla_{\mathcal{W}} J(W)$ with respect to the correlation functional $J(W)$,

$$\nabla_{\mathcal{W}} J(W) = \Pi_W(\nabla J(W)), \qquad J(W) = \langle W, S(W)\rangle, \tag{8}$$

based on the approximation of the Euclidean gradient $\nabla J(W) \approx S(W)$, which is justified by the slow dynamics of $S(W(t))$ due to averaging (7), relative to the fast dynamics of $W(t)$.

### 2.2 Linear Assignment Flow

The *linear assignment flow*, introduced by [16], approximates the mapping (7) as part of the assignment flow (4) by

$$\dot{W} = \Pi_W \Big( S(W_0) + dS_{W_0} \Pi_{W_0} \log \frac{W}{W_0} \Big), \quad W(0) = W_0 = \mathbb{1}_{\mathcal{W}} \in \mathcal{W}. \tag{9}$$

This flow is still *nonlinear* but admits the following parametrization [16, Prop. 4.2]

$$W(t) = \mathrm{Exp}_{W_0}\big(V(t)\big), \quad \dot{V} = \Pi_{W_0}\big(S(W_0) + dS_{W_0} V\big), \quad V(0) = 0, \tag{10}$$

where the latter ODE is *linear* and defined on the tangent space $\mathcal{T}_0$. Assuming that $V$ results from stacking row-wise the tangent vectors $V_i$ for each pixel $i \in I$, the Jacobian $dS_{W_0}$ is given by the block matrix

$$dS_{W_0} = \big(A_{ik}(W_0)\big)_{i,k \in I}, \quad A_{ik}(W_0)(V_k) = \begin{cases} w_{ik} \Pi_{S_i(W_0)}(\frac{V_k}{W_{0k}}), & k \in \mathcal{N}_i, \\ 0, & k \notin \mathcal{N}_i. \end{cases} \tag{11}$$

It is the *linearity* of (10) with respect to both the tangent vector $V$ and the parameters $w_{ik}$ (see (11)), that makes this approach attractive for parameter estimation.

## 3 Learning Adaptive Regularization Parameters

This section describes our contribution as illustrated by Fig. 2: an objective function for determining optimal weights – called *parameters* – that steer the linear assignment flow towards given ground-truth labelings (Sect. 3.1); a *continuous* approach to parameter estimation based on the adjoint dynamical system (Sect. 3.2), which is the method of choice when estimating *many* parameters [5]; alternatively, a *discrete* approach based on discretizing first the parameter estimation problem followed by nonlinear programming (Sect. 3.3); showing that either way yields the *same* result (Sect. 3.4); specifying the resulting algorithm (Sect. 3.5); finally, closing the loop by learning a simple predictor function that maps features extracted from *novel* data to proper weights (Sect. 3.6).

### 3.1 Parameter Estimation by Trajectory Optimization

We consider the following constrained optimization problem

$$\min_{\Omega \in \mathcal{P}} \quad E\big(V(T)\big) \tag{12a}$$

$$\text{s.t.} \quad \dot{V}(t) = f(V(t), \Omega), \quad t \in [0, T], \quad V(0) = 0_{|I| \times |J|}. \tag{12b}$$

where (12b) is given by (10) and the remaining symbols are defined as follows.

$\Omega$ parameters $(w_{ik})_{k\in\mathcal{N}_i}$, $i \in I$ of (11) to be estimated;
$\mathcal{P}$ manifold of parameters $\Omega \in \mathcal{P} := \operatorname{rint} \Delta_{|\mathcal{N}|}^{|I|} = \operatorname{rint}(\Delta_{|\mathcal{N}|} \times \cdots \times \Delta_{|\mathcal{N}|})$;
$|\mathcal{N}|$ equal size of each local neighborhood $|\mathcal{N}| = |\mathcal{N}_i|$ of pixel $i \in I$;
$V(T)$ tangent vectors solving (12b) at termination time $V(t = T)$;
$V(T) = V(T, \Omega)$ depends on $\Omega$ through (12b).

A concrete example for an objective (12a) is

$$E\big(V(T)\big) = D_{\mathrm{KL}}\left(W^*, \exp_{\mathbb{1}_{\mathcal{W}}}\left(V(T)\right)\right), \tag{13}$$

which evaluates the Kullback-Leibler distance of the labeling induced by $V(T)$ from a given ground-truth labeling $W^* \in \mathcal{W}$. In words, our objective is to estimate parameters $\Omega$ that control the linear assignment flow (9), by minimizing $E$. Since the dependency $\Omega \mapsto E$ is only *implicitly* given through (12b), the major task – illustrated by Fig. 2 – is to determine the *sensitivity* $\frac{d}{d\Omega} E(V(T))$, which in turn is used to adapt the parameters. Next, we detail both paths of Figure 2 for evaluating Eq. (1).

### 3.2 Parameter Estimation: Continuous Approach

The following theorem makes precise the upper path to the right of Figure 2.

**Theorem 1 (parameter sensitivity: continuous case).** *Let the objective $E$ be defined by (12). Then*

$$\frac{dE}{d\Omega} = \int_0^T \Big(\frac{\partial f}{\partial \Omega}\Big)^\top \lambda(t) dt, \tag{14a}$$

*where $\lambda(t)$ satisfies the adjoint differential equation*

$$\dot\lambda(t) = -\Big(\frac{\partial f}{\partial V}\Big)^\top \lambda(t), \qquad \lambda(T) = \frac{\partial E}{\partial V}\big(V(T)\big), \tag{14b}$$

*which has to be solved backwards in time.*

*Proof.* Setting up the Lagrangian

$$L(V, \Omega) = E\big(V\big)\big|_{t=T} - \int_0^T \big\langle \lambda, F(\dot V, V, \Omega)\big\rangle dt \tag{15}$$

with multiplier $\lambda(t)$ and $F(\dot V, V, \Omega) := \dot V(t) - f(V(t), \Omega) \equiv 0$, we get

$$\frac{dE}{d\Omega} = \frac{dL}{d\Omega} = \Big(\frac{\partial V}{\partial \Omega}\Big)^\top \frac{\partial E}{\partial V}\Big|_{t=T} - \int_0^T \Big(\frac{\partial F}{\partial \dot V}\frac{\partial \dot V}{\partial \Omega} + \frac{\partial F}{\partial V}\frac{\partial V}{\partial \Omega} + \frac{\partial F}{\partial \Omega}\Big)^\top \lambda \, dt \tag{16}$$

where integration applies componentwise. Using $\frac{\partial F}{\partial \dot V} = I$, we partially integrate the first term under the integral,

$$\int_0^T \Big(\frac{\partial \dot V}{\partial \Omega}\Big)^\top \lambda dt = \Big(\frac{\partial V}{\partial \Omega}\Big)^\top \lambda\Big|_{t=0}^T - \int_0^T \Big(\frac{\partial V}{\partial \Omega}\Big)^\top \dot\lambda \, dt, \tag{17}$$

to obtain with $\frac{\partial F}{\partial \Omega} = -\frac{\partial f}{\partial \Omega}$, $\frac{\partial F}{\partial V} = -\frac{\partial f}{\partial V}$

$$\frac{dE}{d\Omega} = \Big(\frac{\partial V}{\partial \Omega}\Big)^\top \frac{\partial E}{\partial V}\Big|_{t=T} - \Big(\frac{\partial V}{\partial \Omega}\Big)^\top \lambda\Big|_{t=0}^{T} + \int_0^T \Big(\frac{\partial V}{\partial \Omega}\Big)^\top \dot{\lambda}\, dt \tag{18a}$$

$$+ \int_0^T \Big(\frac{\partial f}{\partial V}\frac{\partial V}{\partial \Omega} + \frac{\partial f}{\partial \Omega}\Big)^\top \lambda\, dt. \tag{18b}$$

Thus, with $\frac{\partial V}{\partial \Omega}(0) = 0$, factoring out the unknown Jacobian $\frac{\partial V}{\partial \Omega}$ and choosing $\lambda(t)$ such that the corresponding coefficient vanishes, we obtain

$$\frac{dE}{d\Omega} = \int_0^T \Big(\frac{\partial f}{\partial \Omega}\Big)^\top \lambda(t)\, dt, \quad \text{where } \lambda(t) \text{ solves} \tag{19a}$$

$$\dot{\lambda}(t) = -\Big(\frac{\partial f}{\partial V}\Big)^\top \lambda(t), \qquad \lambda(T) = \frac{\partial E}{\partial V}\big(V(T)\big). \tag{19b}$$

□

**Discretization.** Due to lack of space, we only consider the simplest *geometric* scheme for numerically integrating the ODEs (12b) and (14b),

namely the symplectic Euler method [8], and get

$$V_{k+1} = V_k + h_k f(V_k, \Omega), \tag{20a}$$

$$\lambda_{k+1} = \lambda_k - h_k\big(\partial_V f(V_k, \Omega)\big)^\top \lambda_{k+1}, \tag{20b}$$

with iteration index $k = 0, \dots, N-1$, step sizes $h_k$ and $\sum_{k\in[N]} h_k = T$. This amounts to apply the *explicit* Euler scheme to the linear assignment flow and the *implicit* Euler scheme to the adjoint system. We can avoid the implicit step (20b) by reversing the order of integration $k = N-1, \dots, 0$, to obtain

$$\lambda_k = \lambda_{k+1} + h_k\big(\partial_V f(V_k, \Omega)\big)^\top \lambda_{k+1}, \tag{21}$$

with initial condition $\lambda_N = \lambda(T)$ given by (14b). As a consequence, we first iterate (20a), store all iterates $V_k$ and then iterate (21).

### 3.3 Parameter Estimation: Discrete Approach

In contrast to the previous section, we first *discretize* problem (12) and then *differentiate* the resulting *nonlinear program* (Fig. 2: violet path).

As mentioned in the previous section, we only consider the simplest integration scheme, the explicit Euler method. Using (20a), problem (12) becomes the *nonlinear optimization problem*

$$\min_{\Omega\in\mathcal{P}} \quad E(V_N) \quad \text{s.t.} \quad V_{k+1} = V_k + h_k f(V_k, \Omega), \qquad k = 0, \dots, N-1, \tag{22}$$

with $V_0 = 0_{|I|\times|J|}$. The result analogous to Theorem 1 follows.

**Theorem 2 (parameter sensitivity: discrete case).** *The sensitivity of problem (22) with respect to the parameter $\Omega$ is given by*

$$\frac{dE}{d\Omega} = \sum_{k=1}^{N} h_{k-1}\Big(\frac{\partial f(V_{k-1}, \Omega)}{\partial \Omega}\Big)^\top \overline{\lambda}_k, \tag{23a}$$

*where the discrete adjoint vectors* $\overline{\lambda}_k$ *are given by*

$$\overline{\lambda}_k = \frac{\partial E(V_N)}{\partial V_k}, \quad k = 0, \dots, N. \tag{23b}$$

*Proof.* Skipped due to lack of space. □

We notice that (23a) corresponds to a discretization of (14a). Likewise, the discrete adjoints (23b) can be based on the theory of *automatic differentiation (AD)* [6]. Specifically, in view of the *geometric* numerical integration of the system (20), formula (23a) can be seen as a non-Euclidean version of the *reverse mode* of AD.

### 3.4 Differentiate or Discretize First?

We briefly indicate why either approach of Section 3.2 or Section 3.3 yields the same result, that is diagram of Fig. 2 *commutes* indeed.

We take a closer look at expression (23b) due to the *discrete* problem formulation and compute

$$\overline{\lambda}_k = \frac{\partial E(V_N)}{\partial V_k} = \Big(\frac{\partial V_{k+1}}{\partial V_k}\Big)^\top \frac{\partial E(V_N)}{\partial V_{k+1}} = \Big(\frac{\partial V_{k+1}}{\partial V_k}\Big)^\top \overline{\lambda}_{k+1} \tag{24a}$$

$$\overset{(22)}{=} \Big(\frac{\partial\big(V_k + h_k f(V_k, \Omega)\big)}{\partial V_k}\Big)^\top \overline{\lambda}_{k+1} = \big(I + h_k \partial_V f(V_k, \Omega)\big)^\top \overline{\lambda}_{k+1} \tag{24b}$$

$$\overline{\lambda}_k = \overline{\lambda}_{k+1} + h_k \big(\partial_V f(V_k, \Omega)\big)^\top \overline{\lambda}_{k+1}, \tag{24c}$$

which agrees with (21) derived from the *continuous* problem formulation.

### 3.5 Parameter Estimation Algorithm

Our strategy for *parameter adaption* is to follow the *Riemannian gradient descent flow* on the parameter manifold $\mathcal{P}$,

$$\dot{\Omega} = -\nabla_{\mathcal{P}} E\big(V(T, \Omega)\big) = -\Pi_\Omega\Big(\frac{d}{d\Omega} E\big(V(T, \Omega)\big)\Big), \quad \Omega(0) = \mathbb{1}_{\mathcal{P}}, \tag{25}$$

with $\Pi_\Omega$ given by (5a) and with the *unbiased* initialization $\Omega(0) = \mathbb{1}_{\mathcal{P}}$, i.e. *uniform* weights at every patch $\mathcal{N}_i$ around pixel $i \in I$. We discretize flow (25) using the *geometric* explicit Euler scheme (cf. [16])

$$\Omega_{k+1} = \exp_{\Omega_k}\big(-h_k \nabla_{\mathcal{P}} E(V_N(\Omega_k))\big), \quad \Omega_0 = \Omega(0), \quad k = 1, 2, \dots \tag{26}$$

We summarize the two main procedures for parameter learning.

**Algorithm 1:** Discretized *Riemannian flow.* (25)

**Data:** initial weights $\Omega_0 = \mathbb{1}_{\mathcal{P}}$, objective function $E\big(V(T)\big)$
**Result:** weight parameter estimates $\Omega^*$

```
// geometric Euler integration
for k = 0, ..., K do
    compute (d/dΩ) E(V_N(Ω_k)) ;                    // Algorithm 2
    Ω_{k+1} = exp_{Ω_k}( − h_k Π_Ω( (d/dΩ_k) E(V_N) ));
```

In practice, we terminate the iteration when $\|\Omega_{k+1} - \Omega_k\| \leq \varepsilon$. This determines $K$.

**Algorithm 2:** Compute the sensitivity $\frac{dE(V_N)}{d\Omega}$ by (23a).

**Data:** current weights $\Omega_j$

**Result:** objective value $E(V_N(\Omega_j))$, sensitivity $\frac{dE(V_N)}{d\Omega_j}$

// forward Euler integration

**for** $k = 0, \ldots, N-1$ **do**

  $V_{k+1} = V_k + hf(V_k, \Omega_j)$;

compute $\lambda_N = \frac{\partial E(V_N)}{\partial V_N}$;

set $\frac{dE(V_N)}{d\Omega} = 0$;

// backward Euler integration

**for** $k = N-1, \ldots, 0$ **do**

  $\lambda_k = \lambda_{k+1} + h_k \big(\frac{\partial f(V_k, \Omega_j)}{\partial V}\big)^\top \lambda_{k+1}$;

  $\frac{dE(V_N)}{d\Omega}$ += $h_{k-1} \big(\frac{\partial f(V_{k-1}, \Omega_j)}{\partial \Omega}\big)^\top \lambda_k$ ; // summand of (23a)

### 3.6 Parameter Prediction

We apply kernel-based local regression in order to *generalize* the relation between features and optimal weight parameters for regularization to *novel* data.

Specifically, let $\Omega^* = (w^*_{ik})_{k \in \mathcal{N}_i}$, $i \in I$ denote the output of **Algorithm 1**, and let $\mathcal{F}^*_i = (f^*_k)_{k \in \mathcal{N}_i}$, $i \in I$ denote the corresponding image 'feature patches' of the training data that were used to compute $\Omega^*$. The goal of regression is to generalize the relation between $\mathcal{F}^*_i$, $i \in I$ and $\Omega^*$ to novel image data and features $\mathcal{F}_i$ in terms of a *weight map*

$$\widehat{w} \colon \mathcal{F}_i \to \mathcal{P}, \qquad (f_k)_{k \in \mathcal{N}_i} \to (w_{ik})_{k \in \mathcal{N}_i}, \quad i \in I, \tag{27}$$

where $\widehat{w}$ does *not* depend on the pixel location $i \in I$. In words, the *local spatial context* in terms of novel features $f_k$ observed within an image patch $k \in \mathcal{N}_i$ around each pixel $i \in I$, is used to *predict* the regularization parameters $(w_{ik})_{k \in \mathcal{N}_i}$.

Using the basic Nadaraya–Watson kernel regression estimator [15, Section 5.4] based on the data

$$\big(\mathcal{F}^*_i, (w^*_{ik})_{k \in \mathcal{N}_i}\big)_{i \in N} \tag{28}$$

which constitutes a *coreset* [11] of the data produced offline by **Algorithm 1** from (possibly many) training images, we explore (see Section 4) the predictor map

$$\widehat{w}(\mathcal{F}_i) = \sum_{j \in [N]} \frac{K_h(\mathcal{F}_i, \mathcal{F}^*_j)}{\sum_{j' \in [N]} K_h(\mathcal{F}_i, \mathcal{F}^*_{j'})} w^*_j, \qquad w^*_j = (w^*_{jk})_{k \in \mathcal{N}_j}, \tag{29}$$

where $K_h(\cdot, \cdot)$ is a standard kernel function (Gaussian, Epanechnikov, etc.) applied to a suitable distance between the feature sets $\mathcal{F}_j$ and $\mathcal{F}^*_{j'}$. The *bandwidth parameter* $h$ is determined by cross-validation using the data (28).

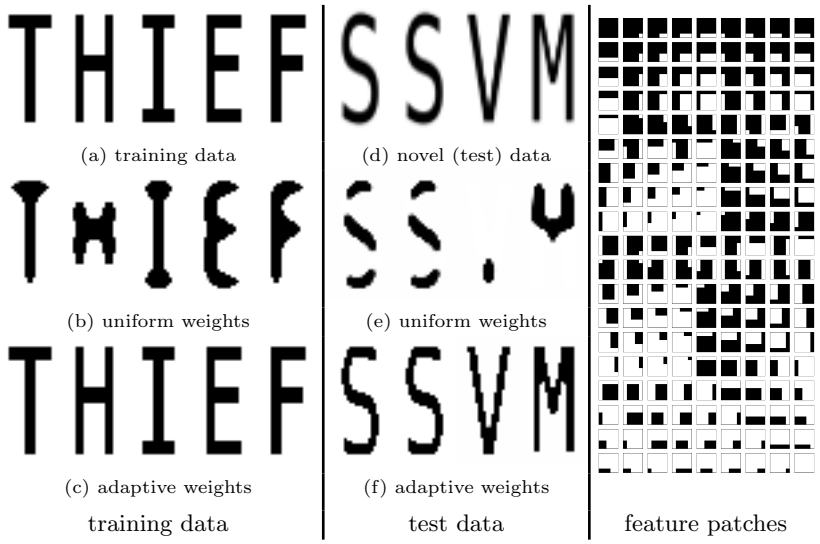



**Fig. 3. Patch-based adaptive regularization of binary letters.** LEFT COLUMN: **(a)** Training data and corresponding $5 \times 5$ feature patches (RIGHTMOST COLUMN). **(b)** Uniform regularization fails even with perfect features. **(c)** Perfect adaptive reconstruction (sanity check). CENTER COLUMN: **(d)** Test data to be labelled using the features and regularizer trained on (a). **(e)** Uniform regularization fails. **(f)** Adaptive regularization predicts *curvilinear* structures of (d) using 'knowledge' based on (a) where *only vertical and horizontal* structures occur.

## 4 Experiments

We illustrate adaptive regularization for image labeling by two further experiments, based on the objective function (13), in addition to the experiment with non-binary data illustrated by Figure 1.

Figure 3 shows patch-based labeling of *curvilinear* letters (center column) using an adaptive regularizer trained on letters with *vertical and horizontal* structures only (left column). See the figure caption for details. This result indicates adaptivity in a two-fold way: use of *non-uniform* weights that are predicted *online* as optimal parameters for *novel* image data *not* seen before.

Figure 4 shows results for *curvilinear line structures* contaminated by noise. The geometry of these scenes correspond to *random* Voronoi diagrams, that is training and test scenes *differ*. The raw grayvalue data together with the outputs of a ridge filter and a Laplacian-of-Gaussian filter were used as feature vectors of dimension 3. The results illustrate the labeling with *uniform* regularization using the smallest scale $|\mathcal{N}| = 3 \times 3$ returns *dilated* and *incomplete* structures, and fails completely at the larger scale $|\mathcal{N}| = 5 \times 5$. The adaptive regularizer yields almost perfect results except for minor boundary effects. This illustrates that after the training phase, when using optimal parameters $\Omega^*$ determined for

training data, then the linear assignment flow has enough predictive power in order to 'know' how to average geometrically when confronted with novel image data not seen before.

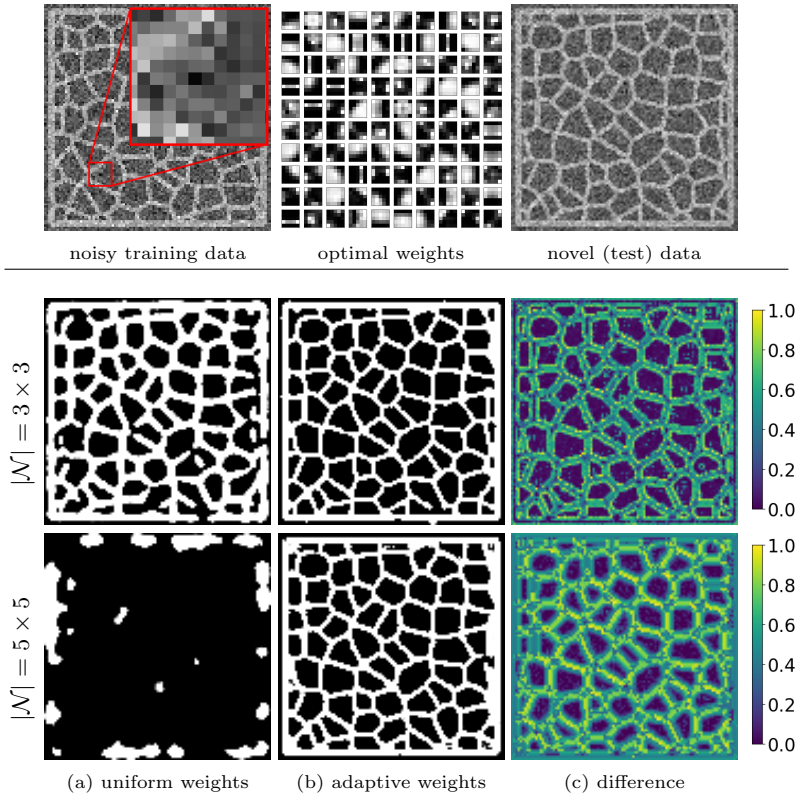



**Fig. 4. Adaptive regularization and labeling of curvilinear noisy line structures.** Training and test data were randomly generated and hence differ completely. Non-adaptive regularization (uniform weights) returns dilated incomplete structures at the smallest scale ($|\mathcal{N}| = 3 \times 3$) or fails completely ($|\mathcal{N}| = 5 \times 5$). Panel 'optimal weights' illustrates a sample of *non-uniform optimal* weight patches $(w^*_{ik})_{k \in \mathcal{N}_i}$, for a couple of pixels $i \in I$, computed during the *training* phase. Panels 'difference' illustrate the deviation of weight patches from uniform weights during the *test* phase for *each* pixel. The corresponding labelings (panels (b)) are almost perfect except for minor boundary effects.

## 5 Conclusion

We introduced an novel approach to image labeling based on adaptive regularization of the linear assignment flow. Parameter estimation relies on a consistent discretization and geometric numerical integration that is easy to reproduce.

The prediction component can be based on any state-of-the-art kernel method from machine learning.

Our future work will extend the numerics to more general symplectic integrators and the approach itself to state-dependent parameter prediction, in connection with spatial multiscale representations of image data.

**Acknowledgement.** Support from the German Science Foundation, grant GRK 1653, is gratefully acknowledged.

## References

1. Amari, S.I., Nagaoka, H.: Methods of Information Geometry. Amer. Math. Soc. and Oxford Univ. Press (2000)
2. Aström, F., Petra, S., Schmitzer, B., Schnörr, C.: Image Labeling by Assignment. J. Math. Imag. Vision **58**(2), 211–238 (2017)
3. Åström, F., Hühnerbein, R., Savarino, F., Recknagel, J., Schnörr, C.: MAP Image Labeling Using Wasserstein Messages and Geometric Assignment. In: Lauze, F., Dong, Y., Dahl, A.B. (eds.) Scale Space and Variational Methods in Computer Vision: 6th International Conference, SSVM 2017, Kolding, Denmark, June 4-8, 2017, Proceedings. pp. 373–385. Springer International Publishing, Cham (2017)
4. E, W.: A Proposal on Machine Learning via Dynamical Systems. Communications in Mathematics and Statistics **5**(1), 1–11 (2017)
5. Giles, M.B., Pierce, N.A.: An Introduction to the Adjoint Approach to Design. Flow, Turbulence and Combustion **65**(3), 393–415 (2000)
6. Griewank, A.: A Mathematical View of Automatic Differentiation. Acta Numerica **12**, 321–398 (2003)
7. Haber, E., Ruthotto, L.: Stable Architectures for Deep Neural Networks. Inverse Problems **34**(1), 014004 (2017)
8. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration. Springer (2006)
9. He, K., Zhang, X., Ren, S., Sun, J.: Deep Residual Learning for Image Recognition. In: Proc. CVPR (2016)
10. Kappes, J., Andres, B., Hamprecht, F., Schnörr, C., Nowozin, S., Batra, D., Kim, S., Kausler, B., Kröger, T., Lellmann, J., Komodakis, N., Savchynskyy, B., Rother, C.: A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. Int. J. Comp. Vision **115**(2), 155–184 (2015)
11. Phillips, J.: Coresets and Sketches. In: Handbook of Discrete and Computational Geometry, chap. 48. CRC Press (2016)
12. Ross, I.: A Roadmap for Optimal Control: The Right Way to Commute. Annals of the New York Academy of Sciences **1065**(1), 210–231 (2018/09/12 2006)
13. Shalev-Shwartz, S., Shamir, O., Shammah, S.: Failures of Gradient-Based Deep Learning. CoRR abs/1703.07950 (2017)
14. Wainwright, M.J.: Estimating the "Wrong" Graphical Model: Benefits in the Computation-Limited Setting. J. Mach. Learning Res. **7**, 1829–1859 (2006)
15. Wasserman, L.: All of Nonparametric Statistics. Springer (2006)
16. Zeilmann, A., Savarino, F., Petra, S., Schnörr, C.: Geometric Numerical Integration of the Assignment Flow. CoRR abs/1810.06970 (2018)
17. Zhu, S.C., Liu, X.: Learning in Gibbsian Fields: How Accurate and How Fast Can It Be? IEEE Trans. Patt. Anal. Mach. Intell. **24**(7), 1001–1006 (2002)