Assignment Flows

Christoph Schnörr

Abstract Assignment flows comprise basic dynamical systems for modeling data labeling and related machine learning tasks in supervised and unsupervised scenarios. They provide adaptive time-variant extensions of established discrete graphical models and a basis for the design and better mathematical understanding of hierarchical networks, using methods from information (differential) geometry, geometric numerical integration, statistical inference, optimal transport and control. This chapter introduces the framework by means of the image labeling problem and outlines directions of current and further research.

1 Introduction

Let $(\mathcal{F}, d_{\mathcal{F}})$ be a metric space and $\mathcal{F}_n = \{f_i \in \mathcal{F} : i \in \mathcal{I}\}$ given data with $|\mathcal{I}| = n$. Assume that a predefined set of *prototypes* $\mathcal{F}_* = \{f_j^* \in \mathcal{F} : j \in \mathcal{J}\}$ is given with $|\mathcal{J}| = c$. *Data labeling* denotes the *assignment*

$$j \to i, \qquad f_j^* \to f_i \tag{1}$$

of a single *prototype* $f_j^* \in \mathcal{F}_*$ to each data point $f_i \in \mathcal{F}_n$. Adopting the common model assumption that $\mathcal{F}_n$ is a *finite sample set* generated by an unknown underlying probability distribution $\mu_{\mathcal{F}}$, the quality of assignments may be defined via the *quantization* of $\mu_{\mathcal{F}}$ in terms of the selected (assigned) prototypes and by corresponding optimality criteria of information theory [16, 18, 14]. The assignment of indices $j \to i$ induces a *partition* (*classification*) of $\mathcal{F}_n$. Accordingly, depending on the research area, prototypes $f_j^* \in \mathcal{F}_*$ are also called *class representatives*, *feature*

C. Schnörr
Heidelberg University, Institute of Applied Mathematics, Im Neuenheimer Feld 205, 69120 Heidelberg, e-mail: schnoerr@math.uni-heidelberg.de

dictionary, *codebook* or simply *labels*, and we will use interchangeably these terms throughout this chapter.

What makes the data labeling problem challenging is that *context-sensitive* label assignments are required: $\mathcal{I}$ forms the vertex set of a given undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ which defines a relation $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$ and neighborhoods

$$\mathcal{N}_i = \{k \in \mathcal{I} : ik \in \mathcal{E}\} \cup \{i\}, \tag{2}$$

where ik is a shorthand for the unordered pair (edge) $(i, k) = (k, i)$. Indices $i \in \mathcal{I}$ frequently index positions $x_i \in \Omega \subset \mathbb{R}^d$ in a Euclidean domain[1], and $k \in \mathcal{N}_i$ indicates a small Euclidean distance $\|x_i - x_k\|$. A basic example are image features $\mathcal{F}_n$ extracted from raw pixel data on a image grid graph $\mathcal{G}$ and the corresponding *image labeling* problem.

In such situations, it is plausible to assume that $k \in \mathcal{N}_i$ implies that the *same* label is assigned to both i and k more frequently than different labels, which explains the success of the total variation measure of 'piecewise image homogeneity' for image denoising [45]. Yet, this assumption falls short of the enormous complexity of *assignment relations* that define *natural real* image structure across the scales up to a semantic level. While information theory clearly says that *joint* assignments are more appropriate than individual assignments for the quantization of complex data sources $\mu_{\mathcal{F}}$ [14], how to accomplish this task in a mathematically and statistically satisfying way using algorithms that are computationally feasible, has remained an unsolved problem.

The aforementioned data encoding-decoding tasks are nowadays mainly performed using deep networks, due to their striking empirical performance in benchmark tests across many disciplines like, e.g., in image labeling [31]. However, this rapid development during recent years has not improved our mathematical understanding in the same way, so far [15]. The 'black box' behavior of deep networks and systematic failures [2] are worrying not only researchers from mathematics and scientific computing, but also industrial partners in connection with safety-critical applications.

In this context, *assignment flows* are introduced in this chapter as an attempt to extend discrete *graphical models* in a systematic way, which defined the prevailing framework for data modeling, inference and learning during the last three decades [17, 33, 51, 30, 36]. Regarding inference *algorithms* using discrete graphical models, we refer to [28] for an assessment of the state of the art.

Assignment flows are *smooth dynamical systems* defined using *information geometry* [1, 4]. Elementary statistical manifolds [32] provide both a target space for *data embedding* and a *state space* on which the assignment flow evolves in order to determine a data labeling. Corresponding vector fields are parametrized and thus enable to learn the *adaptivity* of regularized label assignments within neighborhoods (2), rather than parameters of a *fixed* regularizer as with graphical models or traditional variational approaches to inverse problems. *Smoothness* and *modular compositional*

[1] This includes spatio-temporal data – like e.g. videos – observed at points $(t_i, x_i) \in [0, T] \times \Omega \subset \mathbb{R} \times \mathbb{R}^d$ in time and space.

design yield efficient algorithms based on numerical *geometric integration* and enable to switch seamlessly between supervised and unsupervised scenarios within a single framework.

The assignment flow for *supervised* data labeling is introduced in Section 2. *Unsupervised* scenarios involving *label evolution* and *learning labels from data* are discussed in Section 3. Section 4 reports first steps towards *learning* (*estimating*) adaptivity parameters of the assignment flow via optimal control. Section 5 outlines current and future work that will be undertaken along this research direction, in order to contribute to a better mathematical understanding of the representation and inference of natural image structure.

This chapter focuses on the basic mathematical ingredients and the discussion of corresponding modeling aspects. We refer to [3, 22, 56, 58, 57, 59, 49, 23] for more detailed expositions of the respective topics, numerical experiments and a discussion of related work. Regarding the latter, we include few comments on historical developments as Remarks 1 and 2 on page 7.

Basic Notation. We set $n = |\mathcal{I}|$ (number of vertices), $c = |\mathcal{J}|$ (number of classes resp. labels) and $[m] = \{1, 2, \ldots, m\}$ for $m \in \mathbb{N}$. $\mathbb{1} = (1, 1, \ldots, 1)^\top$ denotes the one-vector whose dimension depends on the context. $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product. The probability simplex of dimension $c - 1$ is

$$\Delta_c = \Big\{ p \in \mathbb{R}^c_+ \colon \langle \mathbb{1}, p \rangle = \sum_{j \in [c]} p^j = 1 \Big\}. \tag{3}$$

It is the convex hull of its vertices (extreme points) which are the unit vectors $e_1 = (1, 0, \ldots, 0)^\top, \ldots, e_c = (0, 0, \ldots, 0, 1)^\top$. The expectation with respect to a distribution $p \in \Delta_c$ is denoted by

$$\mathbb{E}_p[q] = \langle p, q \rangle, \qquad q \in \mathbb{R}^c. \tag{4}$$

$I = \mathrm{Diag}(\mathbb{1})$ denotes the identity matrix.

Inequalities between vectors $\mathbb{R}^c \ni p > 0$ hold for each component, $p^1 > 0, \ldots, p^c > 0$. Likewise, the exponential function and the logarithm apply to each component of the argument vector,

$$e^p = (e^{p^1}, \ldots, e^{p^c})^\top, \qquad \log p = (\log p^1, \ldots, \log p^c)^\top, \tag{5}$$

and componentwise multiplication and subdivision are simply written as

$$uv = (u^1 v^2, \ldots, u^c v^c)^\top, \qquad \frac{v}{p} = \Big(\frac{v^1}{p^1}, \ldots, \frac{v^c}{p^c} \Big)^\top, \qquad u, v \in \mathbb{R}^c, \quad p > 0. \tag{6}$$

It will be convenient to write the exponential function with large expressions as argument in the form $e^p = \exp(p)$. The latter expression should not be confused with the exponential map $\exp_p$ defined by (23) that always involves a subscript. Likewise, log always means the logarithm function and should not be confused with the inverse exponential maps defined by (23).

We write $E(\cdot)$ for specifying various objective functions in this chapter. The context disambiguates this notation.

2 The Assignment Flow for Supervised Data Labeling

We collect in Section 2.1 basic notions of information geometry [32, 1, 10, 4] that are required for introducing the assignment flow for supervised data labeling in Section 2.2. See e.g. [34, 27] regarding general differential geometry and background reading.

2.1 Elements of Information Geometry

We sketch a basic framework of information geometry and then consider the specific instance on which the assignment flow is based.

2.1.1 Dually Flat Statistical Manifolds

Information geometry is generally concerned with smoothly parametrized families of densities on some sample space $\mathcal{X}$ with open parameter set Ξ in a Euclidean space,

$$\Xi \ni \xi \mapsto p(x;\xi), \quad x \in \mathcal{X}, \tag{7}$$

that are regarded as immersions into the space of all densities. Equipped with the Fisher-Rao metric g which is a unique choice due to its invariance against reparametrization,

$$(\mathcal{M}, g) \quad \text{with} \quad \mathcal{M} = \{p(\,\cdot\,;\xi) \colon \xi \in \Xi\} \tag{8}$$

becomes a Riemannian manifold. Let $\mathcal{X}(\mathcal{M})$ denote the space of all smooth vector fields on $\mathcal{M}$. The Riemannian (Levi-Civita) connection ∇^g is the unique affine connection being torsion-free (or symmetric) and compatible with the metric, i.e. the covariant derivative of the metric tensor

$$(\nabla^g_Z g)(X,Y) = 0 \qquad \Leftrightarrow \qquad Zg(X,Y) = g(\nabla^g_Z X, Y) + g(X, \nabla^g_Z Y), \tag{9}$$

vanishes for all $X, Y, Z \in \mathcal{X}(\mathcal{M})$. A key idea of information geometry is to replace ∇^g by two affine connections ∇, ∇^* that are *dual* to each other, which means that they *jointly* satisfy (9),

$$Zg(X,Y) = g(\nabla_Z X, Y) + g(X, \nabla^*_Z Y), \qquad \forall X, Y, Z \in \mathcal{M}(\mathcal{X}). \tag{10}$$

In particular, computations simplify if in addition both ∇ and ∇^* can be chosen *flat*, i.e. for either connection and every point $p_\xi \in \mathcal{M}$ there exists a chart $\mathcal{U} \subset \mathcal{M}$ and local coordinates, called *affine coordinates*, such that the coordinate vector fields are *parallel* in $\mathcal{U}$. $(\mathcal{M}, g, \nabla, \nabla^*)$ is then called a *dually flat statistical manifold*.

2.1.2 The Assignment Manifold

Adopting the framework above, the specific instance of $\mathcal{M}$ relevant to data labeling (classification) is

$$(\mathcal{S}, g), \qquad \mathcal{S} = \{p \in \Delta_c : p > 0\} \tag{11}$$

with sample space $\mathcal{J} = [c]$,

$$\mathbb{1}_{\mathcal{S}} = \frac{1}{c}\mathbb{1} \in \mathcal{S}, \qquad \text{(barycenter)} \tag{12}$$

tangent bundle $T\mathcal{S} = \mathcal{S} \times T_0$,

$$T_0 = \{v \in \mathbb{R}^c : \langle \mathbb{1}, v \rangle = 0\}, \tag{13}$$

orthogonal projection

$$\Pi_0 : \mathbb{R}^c \to T_0, \qquad \Pi_0 = I - \mathbb{1}_{\mathcal{S}}\mathbb{1}^\top, \tag{14}$$

and Fisher-Rao metric

$$g_p(u, v) = \sum_{j \in \mathcal{J}} \frac{u^j v^j}{p^j}, \quad p \in \mathcal{S}, \quad u, v \in T_0. \tag{15}$$

Given a smooth function $f : \mathbb{R}^c \to \mathbb{R}$ and its restriction to $\mathcal{S}$, also denoted by f, with Euclidean gradient ∂f_p,

$$\partial f_p = (\partial_1 f_p, \ldots, \partial_c f_p)^\top, \tag{16}$$

the Riemannian gradient reads

$$\operatorname{grad}_p f = R_p \partial f_p = p\big(\partial f_p - \mathbb{E}_p[\partial f_p]\mathbb{1}\big) \tag{17}$$

with the linear map

$$R_p : \mathbb{R}^c \to T_0, \qquad R_p = \operatorname{Diag}(p) - pp^\top, \qquad p \in \mathcal{S} \tag{18}$$

satisfying

$$R_p = R_p \Pi_0 = \Pi_0 R_p. \tag{19}$$

The affine connections ∇, ∇^* are flat and given by the *e-connection* and *m-connection*, respectively, where 'e' and 'm' stand for the exponential and mixture

representation of distributions $p \in \mathcal{S}$ [1]. The corresponding affine coordinates are given by $\theta \in \mathbb{R}^{c-1}$ and $0 < \mu \in \mathbb{R}^{c-1}$ with $\langle \mathbb{1}, \mu \rangle < 1$ such that

$$p = p_\theta = \frac{1}{1 + \langle \mathbb{1}, e^\theta \rangle} (e^{\theta^1}, \ldots, e^{\theta^{c-1}}, 1)^\top \in \mathcal{S}, \tag{20a}$$

$$p = p_\mu = (\mu^1, \ldots, \mu^{c-1}, 1 - \langle \mathbb{1}, \mu \rangle)^\top \in \mathcal{S}. \tag{20b}$$

Choosing affine geodesics in the parameter spaces

$$\theta(t) = \theta + t\dot{\theta}, \qquad \mu(t) = \mu + t\dot{\mu}, \tag{21}$$

the affine e- and m-geodesics in $\mathcal{S}$ read with $p = p_\theta = p_\mu \in \mathcal{S}$

$$p_{\theta(t)} = \frac{p e^{t \frac{v}{p}}}{\langle p, e^{t \frac{v}{p}} \rangle}, \qquad v = \begin{pmatrix} \dot{\mu} \\ -\langle \mathbb{1}, \dot{\mu} \rangle \end{pmatrix} \in T_0, \tag{22a}$$

$$p_{\mu(t)} = p_\mu + tv, \qquad t \in [t_{\min}, t_{\max}], \tag{22b}$$

where t in (22b) has to be restricted to an interval around $0 \in [t_{\min}, t_{\max}]$, depending on μ and v, such that $p_{\mu(t)} \in \mathcal{S}$. Therefore, regarding numerical computations, it is more convenient to work with the *unconstrained* e-representation.

Accordingly, with $v \in T_0$, $p \in \mathcal{S}$, we define the exponential maps and their inverses

$$\mathrm{Exp} \colon \mathcal{S} \times T_0 \to \mathcal{S}, \qquad (p, v) \mapsto \mathrm{Exp}_p(v) = \frac{p e^{\frac{v}{p}}}{\langle p, e^{\frac{v}{p}} \rangle}, \tag{23a}$$

$$\mathrm{Exp}_p^{-1} \colon \mathcal{S} \to T_0, \qquad q \mapsto \mathrm{Exp}_p^{-1}(q) = R_p \log \frac{q}{p}, \tag{23b}$$

$$\exp_p \colon T_0 \to \mathcal{S}, \qquad \exp_p = \mathrm{Exp}_p \circ R_p, \tag{23c}$$

$$\exp_p^{-1} \colon \mathcal{S} \to T_0, \qquad \exp_p^{-1}(q) = \Pi_0 \log \frac{q}{p}, \tag{23d}$$

Applying the map $\exp_p$ to a vector in $\mathbb{R}^c = T_0 \oplus \mathbb{R}\mathbb{1}$ does not depend on the constant component of the argument, due to (19).

The **assignment manifold** is defined as

$$(\mathcal{W}, g), \qquad \mathcal{W} = \mathcal{S} \times \cdots \times \mathcal{S}. \qquad (n = |\mathcal{I}| \text{ factors}) \tag{24}$$

Points $W \in \mathcal{W}$ are row-stochastic matrices $W \in \mathbb{R}^{n \times c}$ with row vectors

$$W_i \in \mathcal{S}, \quad i \in \mathcal{I} \tag{25}$$

that represent the assignments (1) for every $i \in \mathcal{I}$. The jth component of W_i is interchangeably denoted by W_i^j or as element $W_{i,j}$ of the matrix $W \in \mathcal{W}$.

We set

$$\mathcal{T}_0 = T_0 \times \cdots \times T_0 \qquad (n = |\mathcal{I}| \text{ factors}) \tag{26}$$

with tangent vectors $V \in \mathbb{R}^{n\times c},\ V_i \in T_0,\ i \in \mathcal{I}$. All the mappings defined above factorize in a natural way and apply row-wise, e.g. $\mathrm{Exp}_W = (\mathrm{Exp}_{W_1}, \ldots, \mathrm{Exp}_{W_n})$ etc.

Remark 1 (Early related work: nonlinear relaxation labeling)

Regarding *image labeling*, our work originates in the seminal work of Rosenfeld, Hummel and Zucker [43, 24]. Similar to the early days of neural networks [46], this approach was not accepted by researchers focusing on applications. Rather, support vector machines [13] were prevailing later on in pattern recognition and machine learning due to the convexity of the training problem, whereas graph cuts [9] became the workhorse for image labeling (segmentation), for similar reasons.

Nowadays, deep networks predominate in any field due to its unprecedented performance in applications. And most practitioners, therefore, accept it and ignore the criticism in the past that has not become obsolete [15, 2].

Remark 2 (Related work: replicator equation and evolutionary game dynamics)

The gradient flow

$$\dot{p} = \mathrm{grad}_p\, f, \qquad p(0) \in p_0 \in \mathcal{S} \tag{27}$$

evolving on $\mathcal{S}$, with $\mathrm{grad}_p\, f$ due to (17), is known as the *replicator equation* in connection with evolutionary dynamical games [21, 47]. More general 'payoff functions' replacing ∂f_p in (17) have been considered that may or may not derive from a potential.

In view of Remark 1, we point out that Pelillo [39] worked out connections to relaxation labeling from this angle and, later on, also to graph-based clustering [38]. In our opinion, a major reason for why these approaches fall short of the performance of alternative schemes is the absence of a spatial interaction mechanism that conforms with the underlying geometry of assignments. Such a mechanism basically defines the assignment flow to be introduced below.

2.2 The Assignment Flow

We introduce the assignment flow [3] and its components for supervised data labeling on a graph.

2.2.1 Likelihood Map

Let $i \in \mathcal{I}$ be any vertex and (recall (1))

$$D_i = \big(d_{\mathcal{F}}(f_i, f_1^*), \ldots, d_{\mathcal{F}}(f_i, f_c^*)\big)^\top, \qquad i \in \mathcal{I}. \tag{28}$$

Since the metric (feature) space $\mathcal{F}$ can be anything depending on the application at hand, we include a scaling parameter[2] $\rho > 0$ for normalizing the range of the components of D_i and define the **likelihood map** in terms of the **likelihood vectors**

$$L_i : \mathcal{S} \to \mathcal{S}, \qquad L_i(W_i) = \exp_{W_i}\Big(-\frac{1}{\rho} D_i\Big) = \frac{W_i e^{-\frac{1}{\rho}D_i}}{\langle W_i, e^{-\frac{1}{\rho}D_i}\rangle}, \qquad i \in \mathcal{I}. \tag{29}$$

By (23c) and (17), a likelihood vector (29) is formed by regarding D_i as gradient vector (see also Remark 3 below) and applying the exponential map Exp_{W_i}.

Using (29) we define the **single-vertex assignment flow**

$$\dot{W}_i = R_{W_i} L_i(W_i), \quad W_i(0) = \mathbb{1}_{\mathcal{S}} \tag{30a}$$

$$= W_i\big(L_i(W_i) - \mathbb{E}_{W_i}[L_i(W_i)]\mathbb{1}\big), \quad i \in \mathcal{I}. \tag{30b}$$

We have

Proposition 1 *The solution to the system* (30) *satisfies*

$$\lim_{t\to\infty} W_i(t) = W_i^* = \frac{1}{|J_*|}\sum_{j\in J_*} e_j \in \arg\min_{W_i\in\Delta_c} \langle W_i, D_i\rangle, \quad J_* = \arg\min_{j\in J} D_i^j. \tag{31}$$

In particular, if the distance vector D_i has a unique minimal component $D_i^{j_*}$, then $\lim_{t\to\infty} W_i(t) = e_{j_*}$.

Remark 3 (Data term, variational continuous cuts)

A way to look at (29) that has proven to be useful for generalizations of the assignment flow (cf. Section 3.2), is to regard D_i as Euclidean gradient of the *data term*

$$W_i \mapsto \langle D_i, W_i\rangle \tag{32}$$

of established variational approaches ('continuous cuts') to image labeling, cf. [35, Eq. (1.2)] and [11, Thm. 2] for the specific binary case of $c = 2$ labels. Minimizing this data term over $W_i \in \Delta_c$ yields the result (31). In this sense, (29) and (30) provide a *smooth geometric* version of traditional data terms of variational approaches to data labeling and a dynamic 'local rounding' mechanism, respectively.

2.2.2 Similarity Map

The flow (30) does not interact with the flow at any other vertex $i' \in \mathcal{I}$. In order to couple these flows within each neighborhood $\mathcal{N}_i$ given by (2), we assign to each

[2] The sizes of the components D_i^j, $j \in \mathcal{J}$ *relative* to each other only matter.

such neighborhood the positive weights[3]

$$\Omega_i = \Big\{ w_{i,k} \colon k \in \mathcal{N}_i,\ w_{i,k} > 0,\ \sum_{k \in \mathcal{N}_i} w_{i,k} = 1 \Big\}, \quad i \in \mathcal{I} \tag{33}$$

and define the **similarity map** in terms of the **similarity vectors**

$$S_i \colon \mathcal{W} \to \mathcal{S}, \qquad S_i(W) = \exp_{W_i} \Big(\sum_{k \in \mathcal{N}_i} w_{i,k} \exp_{W_i}^{-1} \big(L_k(W_k) \big) \Big) \tag{34a}$$

$$= \frac{\prod_{k \in \mathcal{N}_i} L_k(W_k)^{w_{i,k}}}{\big\langle \mathbb{1}, \prod_{k \in \mathcal{N}_i} L_k(W_k)^{w_{i,k}} \big\rangle}, \quad i \in \mathcal{I}. \tag{34b}$$

The meaning of this map is easy to see: The argument of (34a) in round brackets corresponds to the optimality condition that determines the Riemannian mean of the likelihood vectors L_k, $k \in \mathcal{N}_i$ with respect to the discrete measure Ω_i, if the exponential map of the Riemannian connection were used [27, Lemma 6.9.4]. Using instead the exponential map of the e-connection yields the closed-form formula (34b) that can be computed efficiently.

Remark 4 (Parameters)

Two parameters have been introduced at this point: the *size* $|\mathcal{N}_i|$ of the neighborhoods (2) that we regard as a *scale parameter*, and the *weights* (33). How to turn the weights in *adaptivity parameters* and to learn them from data is discussed in Section 4.

2.2.3 Assignment Flow

The interaction of the single-vertex flows through the similarity map defines the **assignment flow**

$$\dot{W} = R_W S(W), \qquad W(0) = \mathbb{1}_{\mathcal{W}}, \tag{35a}$$

$$\dot{W}_i = R_{W_i} S_i(W), \qquad W_i(0) = \mathbb{1}_{\mathcal{S}}, \quad i \in \mathcal{I}, \tag{35b}$$

where $\mathbb{1}_{\mathcal{W}} \in \mathcal{W}$ denotes the barycenter of $\mathcal{W}$, each row of which is equal to $\mathbb{1}_{\mathcal{S}}$. System (35a) collects the local systems (35b), for each $i \in \mathcal{I}$, which are coupled through the neighborhoods $\mathcal{N}_i$ and the similarity map (34).

Observe the structural similarity of (30a) and (35) due to the *composition* of the likelihood and similarity maps, unlike the traditional *additive* combination of data and regularization terms.

[3] Here we overload the symbol Ω which denotes the Euclidean domain covered by the graph $\mathcal{G}$, as mentioned after Eq. (2). Due to the subscripts Ω_i and the context, there should be no danger of confusion.

Example. Consider the case of two vertices $\mathcal{I} = \{1, 2\}$ and two labels $c = 2$. Parametrize the similarity vectors by

$$S_1 = (s_1, 1 - s_1)^\top, \qquad S_2 = (s_2, 1 - s_2)^\top, \qquad s_i \in (0, 1), \qquad i \in \mathcal{I} \tag{36a}$$

and the weights $\Omega_i = \{w_{i,1}, w_{i,2}\}$ by

$$w_{11} = w_1, \quad w_{12} = 1 - w_1, \quad w_{21} = 1 - w_2, \quad w_{22} = w_2, \qquad w_i \in (0, 1) \tag{36b}$$

for $i \in \mathcal{I}$. Due to this parametrization, one can show that the assignment flow for this special case is essentially governed by the system

$$\begin{pmatrix} \dot{s}_1 \\ \dot{s}_2 \end{pmatrix} = \begin{pmatrix} s_1(1 - s_1)\big(w_1(2s_1 - 1) + (1 - w_1)(2s_2 - 1)\big) \\ s_2(1 - s_2)\big((1 - w_2)(2s_1 - 1) + w_2(2s_2 - 1)\big) \end{pmatrix} \tag{37}$$

with initial values $s_1(0), s_2(0)$ depending on the data D_1, D_2. Figure 1 illustrates that the weights control the stability of stationary points at the extreme points that correspond to unambiguous labelings to which the assignment flow may converge, and the regions of attraction. Interior fixed points exist as well, including interior points of the facets, but are unstable.

A corresponding study of the general case will be reported in future work.

2.2.4 Geometric Integration

We numerically compute the assignment flow by *geometric integration* of the system of ODEs (35). Among a range of possible methods [19], Lie group methods [26] are particularly convenient if they can be applied. This requires to point out a Lie group G and an action $\Lambda\colon \mathrm{G} \times \mathcal{M} \to \mathcal{M}$ of G on the manifold $\mathcal{M}$ at hand such that the ODE to be integrated can be represented by a corresponding Lie algebra action [26, Assumption 2.1].

In the case of the assignment flow, we simply identify $\mathrm{G} = T_0$ with the *flat* tangent space. One easily verifies that the action $\Lambda\colon T_0 \times \mathcal{S} \to \mathcal{S}$ defined as

$$\Lambda(v, p) = \exp_p(v), \tag{38}$$

satisfies

$$\Lambda(0, p) = p, \tag{39a}$$

$$\Lambda(v_1 + v_2, p) = \frac{p e^{v_1 + v_2}}{\langle p, e^{v_1 + v_2} \rangle} = \Lambda(v_1, \Lambda(v_2, p)). \tag{39b}$$

Based on Λ the 'Lie machinery' can be applied [56, Section 3] and eventually leads to the following tangent space parametrization of the assignment flow.

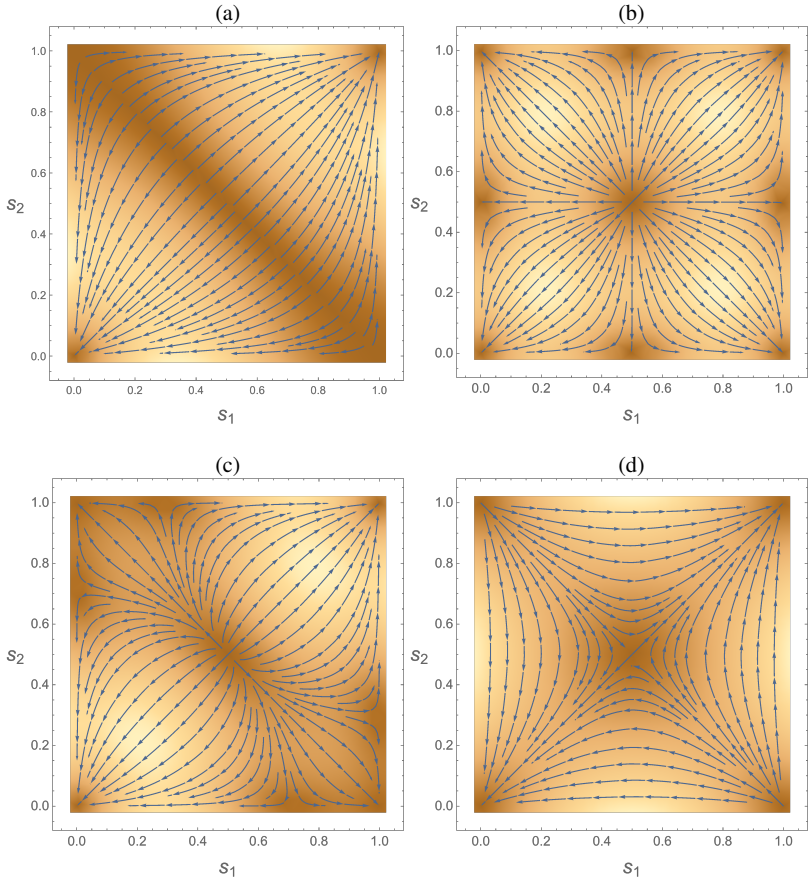

Fig. 1 Vector field on the right-hand side of the ODE-system (37) that represents the assignment flow for two vertices and two labels, for different weight values w_1, w_2. Depending on these weights, we observe stable and unstable stationary points at the vertices that represent the possible labelings, and throughout unstable interior stationary points (including interior points of the facets) that correspond to ambiguous labelings.

Proposition 2 ([56]) *The solution $W(t)$ to assignment flow* (35) *emanating from any $W_0 = W(0)$ admits the representation*

$$W(t) = \exp_{W_0}\big(V(t)\big) \tag{40a}$$

where $V(t) \in \mathcal{T}_0$ solves

$$\dot{V} = \Pi_{\mathcal{T}_0} S\big(\exp_{W_0}(V)\big), \qquad V(0) = 0 \tag{40b}$$

and $\Pi_{\mathcal{T}_0}$ *denotes the natural extension of the orthogonal projection* (14) *onto the tangent space* (26).

We refer to [56] for an evaluation of geometric RKMK methods [37] including embedding schemes for adaptive stepsize selection and more. These algorithms efficiently integrate not only the basic assignment flow but also more involved extensions to unsupervised scenarios, as discussed in Section 3.

2.2.5 Evaluation of Discrete Graphical Models: Wasserstein Message Passing

In Section 1 the assignment flow was motivated and characterized as an approach that extends discrete graphical models in a systematic way. A natural question, therefore, is: How can one *evaluate* a *given* graphical model using the assignment flow?

This problem was studied in [22]. Let

$$\ell \colon \mathcal{I} \to J \tag{41}$$

denote a labeling variable defined on the graph $\mathcal{G}$. We regard ℓ both as a *function* $\mathcal{I} \ni i \mapsto \ell_i \in \mathcal{J}$ and as a *vector* $\mathcal{I} \ni i \mapsto \ell_i = (\ell_i^1, \ldots, \ell_i^c)^\top \in \{e_1, \ldots, e_{|\mathcal{J}|}\}$ depending on the context.

The basic MAP-inference problem (MAP = maximum a posteriori) amounts to minimize a given discrete energy function with arbitrary local functions E_i, E_{ik},

$$E(\ell) = \sum_{i \in \mathcal{I}} E_i(\ell_i) + \sum_{ik \in \mathcal{E}} E_{ik}(\ell_i, \ell_k), \tag{42}$$

which is a combinatorially hard problem. The local interaction functions are typically specified in terms of a metric $d_{\mathcal{J}}$ of the label space $(\mathcal{J}, d_{\mathcal{J}})$,

$$E_{ik}(\ell_i, \ell_k) = d_{\mathcal{J}}(\ell_i, \ell_k), \tag{43}$$

in which case the problem to minimize $E(\ell)$ is also called the *metric labeling problem* [29]. The basic idea of the approach [22] is

(a) to rewrite the local energy terms in the form

$$E(\ell) = \sum_{i \in \mathcal{I}} \Big(\langle \theta_i, \ell_i \rangle + \frac{1}{2} \sum_{k \in \mathcal{N}_i} \langle \ell_i, \Theta_{ik} \ell_k \rangle \Big) \tag{44}$$

with local parameter vectors θ_i and matrices Θ_{ik} given by

$$\langle \theta_i, e_j \rangle = E_i(j), \qquad \langle e_j, \Theta_{ik} e_{j'} \rangle = d_{\mathcal{J}}(j, j'), \qquad i, k \in \mathcal{I}, \quad j, j' \in \mathcal{J}; \tag{45}$$

(b) to define the energy function (42) on the assignment manifold by substituting assignment variables for the labeling variables,

$$\ell \to W \in \mathcal{W}, \qquad \ell_i \to W_i \in \mathcal{S}, \qquad i \in \mathcal{I}; \tag{46}$$

this constitutes a problem *relaxation*;

(c) to turn the interaction term into *smoothed* local Wasserstein distances

$$d_{\Theta_{ik},\tau}(W_i, W_k) = \min_{W_{ik} \in \Pi(W_i, W_k)} \left\{ \langle \Theta_{ik}, W_{ik} \rangle + \tau \sum_{j,j' \in \mathcal{J}} W_{ik,jj'} \log W_{ik,jj'} \right\} \quad (47a)$$

$$\text{subject to} \quad W_{ik}\mathbb{1} = W_i, \quad W_{ik}^\top \mathbb{1} = W_k \quad (47b)$$

between the assignment vectors considered as local marginal measures and using Θ_{ik} as costs for the 'label transport'. Problem (47) is a linear assignment problem regularized by the negative entropy which can be efficiently solved by iterative scaling of the coupling matrix W_{ik} [25].

As a result, one obtains the relaxed energy function

$$E_\tau(W) = \sum_{i \in \mathcal{I}} \left(\langle \theta_i, W_i \rangle + \frac{1}{2} \sum_{k \in \mathcal{N}_i} d_{\Theta_{ik},\tau}(W_i, W_k) \right) \quad (48)$$

with smoothing parameter $\tau > 0$, that properly takes into account the interaction component of the graphical model.

Objective function (48) is continuously differentiable. Replacing D_i in the likelihood map (29) by $\partial_{W_i} E_\tau(W)$ – cf. the line of reasoning mentioned as Remark 3 – bases the likelihood map on *state-dependent distances* that take into account the interaction of label assignment with the neighborhoods $\mathcal{N}_i$ of the underlying graph, as specified by the given graphical model. This regularizing component of $\partial_{W_i} E_\tau(W)$ replaces the geometric averaging (34). An entropy term $\alpha H(W)$ is added in order to gradually enforce an integral assignment W. Numerical integration yields $W(t)$ which converges to a local minimum of the *discrete* objective function (42) whose quality (energy value) depends on the tradeoff – controlled by the single parameter α – between minimizing the relaxed objective function (48) and approaching an integral solution (unambiguous labeling).

The corresponding 'data flow' along the edges of the underlying graph resembles established belief propagation algorithms [55], yet with significant conceptual differences. For example, the so-called local-polytope constraints of the standard polyhedral relaxation of discrete graphical models (cf. Section 2.2.6) are satisfied *throughout* the iterative algorithm, rather than *after* convergence only. This holds by construction due to the 'Wasserstein messages' the result from the local Wasserstein distances of (48), once the partial gradients $\partial_{W_i} E_\tau(W)$, $i \in \mathcal{I}$ are computed. We refer to [22] for further details and discussion.

2.2.6 Global Static vs. Local Dynamically Interacting Statistical Models

The standard polyhedral convex relaxation [54] of the discrete optimization problem (42) utilizes a *linearization* of (44), rewritten in the form

$$E(\ell) = \sum_{i \in \mathcal{I}} \langle \theta_i, \ell_i \rangle + \sum_{ik \in \mathcal{E}} \sum_{j,j' \in \mathcal{J}} \Theta_{ik,jj'} \ell_i^j \ell_k^{j'}, \tag{49}$$

by introducing auxiliary variables $\ell_{ik,jj'}$ that replace the quadratic terms $\ell_i^j \ell_k^{j'}$. Collecting all variables $\ell_i^j, \ell_{ik,jj'},\ i,k \in \mathcal{I},\ j,j' \in \mathcal{J}$ into vectors $\ell_{\mathcal{I}}$ and $\ell_{\mathcal{E}}$ and similarly for the model parameters to obtain vectors $\theta_{\mathcal{I}}$ and $\theta_{\mathcal{E}}$, enables to write (49) as linear form

$$E(\ell) = \langle \theta_{\mathcal{I}}, \ell_{\mathcal{I}} \rangle + \langle \theta_{\mathcal{E}}, \ell_{\mathcal{E}} \rangle, \qquad \ell = (\ell_{\mathcal{I}}, \ell_{\mathcal{E}}) \tag{50}$$

and to define the probability distribution

$$p(\ell; \theta) = \exp\big(\langle \theta_{\mathcal{I}}, \ell_{\mathcal{I}} \rangle + \langle \theta_{\mathcal{E}}, \ell_{\mathcal{E}} \rangle - \psi(\theta) \big), \tag{51}$$

which is a member of the exponential family of distributions [5, 51]. $p(\ell;\theta)$ is the *discrete graphical model* corresponding to the discrete energy function (42) with log-partition function

$$\psi(\theta) = \log \sum_{\ell \in \text{labelings}} \exp\big(\langle \theta_{\mathcal{I}}, \ell_{\mathcal{I}} \rangle + \langle \theta_{\mathcal{E}}, \ell_{\mathcal{E}} \rangle \big). \tag{52}$$

The aforementioned polyhedral convex relaxation is based on the substitution (46) and replacing the integrality constraints on ℓ by

$$W_i \in \Delta_c, \quad i \in \mathcal{I} \tag{53a}$$

and further affine constraints

$$\sum_{j \in \mathcal{J}} W_{ik,jj'} = W_{k,j'}, \qquad \sum_{j' \in \mathcal{J}} W_{ik,jj'} = W_{i,j}, \qquad \forall i,k \in \mathcal{I},\ \forall j,j' \in \mathcal{J} \tag{53b}$$

that ensure local consistency of the linearization step from (49) to (50). While this so-called *local polytope relaxation* enables to compute good suboptimal minima of (42) by solving a (typically huge) linear program as defined by (50) and (53) using dedicated solvers [28], it has also a major mathematical consequence: the graphical model (51) is *overcomplete* or *non-minimally represented* [51] due to linear dependencies among the constraints (53b). For this reason the model (51) cannot be regarded as point on a *smooth* statistical *manifold* as outlined in Section 2.1.1.

In this context, the assignment flow may be considered as an approach that emerges from an antipodal starting point. Rather than focusing on the *static global* and *overcomplete* model of the exponential family (51) defined on the entire graph $\mathcal{G}$, we assign to each vertex $i \in \mathcal{I}$ a discrete distribution $W_i = (W_i^1, \ldots, W_i^c)^\top$, which by means of the parametrization (20a) can be recognized as *minimally represented* member of the exponential family

$$W_i^{\ell_i} = p(\ell_i;\theta_i) = \exp\big(\langle(\theta_i, 1), e_{\ell_i}\rangle - \psi(\theta_i)\big), \quad \ell_i \in \mathcal{J}, \quad i \in \mathcal{I} \tag{54a}$$

$$\psi(\theta_i) = \log(1 + \langle \mathbb{1}, e^{\theta_i}\rangle), \tag{54b}$$

and hence as point $W_i \in \mathcal{S}$ of the statistical manifold $\mathcal{S}$. These states of label assignments *dynamically* interact through the *smooth* assignment flow (35).

We point out that the parameters θ_i of (54) are the affine coordinates of $\mathcal{S}$ and have nothing to do with the model parameter $\theta_{\mathcal{I}}, \theta_{\mathcal{E}}$ of the graphical model (51). The counter part of $\theta_{\mathcal{I}}$ are the distance vectors D_i, $i \in \mathcal{I}$ (28) as part of the likelihood map (29), whereas the counterpart of $\theta_{\mathcal{E}}$ are the weights Ω_i, $i \in \mathcal{I}$ (33) as part of the similarity map (34). The parameters $\theta_{\mathcal{I}}, \theta_{\mathcal{E}}$ are *static (fixed)*, whereas the smooth geometric setting of the assignment flow facilitates computationally the *adaption* of D_i, Ω_i, $i \in \mathcal{I}$. Examples for the adaption of distances D_i are the state-dependent distances discussed in Section 2.2.5 (cf. the paragraph after Eq. (48)) and in the unsupervised scenario of Section 3.2. Adapting the weights Ω_i by learning from data is discussed in Section 4.

Regarding numerical computations, using discrete graphical models to cope with such tasks is more cumbersome.

3 Unsupervised Assignment Flow and Self-Assignment

Two extensions of the assignment flow to unsupervised scenarios are considered in this section. The ability to adapt labels on a feature manifold, during the evolution of the assignment flow, defines the *unsupervised assignment flow* [58, 57] introduced in Section 3.1. On the other hand, learning labels directly from data without any prior information defines the *self-assignment flow* [59] introduced in Section 3.2.

3.1 Unsupervised Assignment Flow: Label Evolution

Specifying a proper set $\mathcal{F}_*$ of labels (prototypes) beforehand is often difficult in practice: Determining prototypes by clustering results in suboptimal quantizations of the underlying feature space $\mathcal{F}_n$. And carrying out this task without the context that is required for proper inference (label assignment) makes the problem ill-posed, to some extent.

In order to alleviate this issue, a natural approach is to *adapt* an initial label set *during* the evolution of the assignment flow. This is done by coupling label and assignment evolution with interaction in both directions: labels define a time-variant distance vector field that steers the assignment flow, whereas regularized assignments move labels to proper positions in the feature space $\mathcal{F}$.

In this section, we make the stronger assumption that $(\mathcal{F}, g_{\mathcal{F}})$ is a smooth Riemannian feature manifold with metric $g_{\mathcal{F}}$. The corresponding linear tangent-cotangent isomorphism $\widehat{g}_{\mathcal{F}}$ connecting differentials and gradients of smooth func-

tions $f\colon \mathcal{F} \to \mathbb{R}$ is given by

$$\operatorname{grad} f = \widehat{g_{\mathcal{F}}}^{-1}(df). \tag{55}$$

Furthermore, we assume a smooth divergence function [6] to be given,

$$D_{\mathcal{F}}\colon \mathcal{F} \times \mathcal{F} \to \mathbb{R}, \qquad D_{\mathcal{F}}(f, f') \approx \frac{1}{2} d_{\mathcal{F}}(f, f')^2, \tag{56}$$

that approximates the squared Riemannian distance, including equality as special case. A proper choice of $D_{\mathcal{F}}$ is crucial for applications: It ensures that approximate Riemannian means can be computed efficiently. See [57, Section 5] for few scenarios worked out in detail.

Let

$$F^*(t) = \{f_1^*(t), \ldots, f_c^*(t)\}, \qquad t \in [0, T] \tag{57}$$

denote set of evolving feature prototypes, with initial set $F^*(0) = F_0^*$ computed by any efficient method like metric clustering [20], and with the final set $F^*(T) = \mathcal{F}_*$ of *adapted* prototypes. In order to determine $F^*(t)$, the assignment flow (35) is extended to the system

$$\dot{F}^* = \mathcal{V}_{\mathcal{F}}(W, F^*), \qquad F^*(0) = F_0^*, \tag{58a}$$

$$\dot{W} = \mathcal{V}_{\mathcal{W}}(W, F^*), \qquad W(0) = \mathbb{1}_{\mathcal{W}}. \tag{58b}$$

The solution $F^*(t)$ to (58a) evolves on the feature manifold $\mathcal{F}$. It is driven by local Riemannian means that are regularized by the assignments $W(t)$. Equation (58b) is the assignment flow determining $W(t)$, based on a time-variant distance vector field in the likelihood map (29) due to the moving labels $F^*(t)$.

A specific formulation of (58) is worked out in [57] in terms of a one-parameter family of vector fields $(\mathcal{V}_{\mathcal{F}}, \mathcal{V}_{\mathcal{W}})$ that define the following **unsupervised assignment flow** for given data $\mathcal{F}_n = \{f_1, \ldots, f_n\}$,

$$\dot{f}_j^* = -\alpha \sum_{i \in \mathcal{I}} \nu_{j|i}(W, F^*) \widehat{g_{\mathcal{F}}}^{-1}\big(d_2 D_{\mathcal{F}}(f_i, f_j^*)\big), \qquad f_j^*(0) = f_{0,j}^*, \quad j \in \mathcal{J}, \tag{59a}$$

$$\dot{W}_i = R_{W_i} S_i(W), \qquad W_i(0) = \mathbb{1}_{\mathcal{S}}, \quad i \in \mathcal{I}, \tag{59b}$$

with parameter $\alpha > 0$ controlling the speed of label vs. assignment evolution, and

$$\nu_{j|i}(W, F^*) = \frac{L_{i,j}^{\sigma}(W, F^*)}{\sum_{k \in \mathcal{I}} L_{k,j}^{\sigma}(W, F^*)}, \qquad L_{i,j}^{\sigma}(W, F^*) = \frac{W_{i,j} e^{-\frac{1}{\sigma} D_{\mathcal{F}}(f_i, f_j^*)}}{\sum_{l \in \mathcal{J}} W_{i,l} e^{-\frac{1}{\sigma} D_{\mathcal{F}}(f_i, f_l^*)}} \tag{60}$$

with parameter $\sigma > 0$ that smoothly 'interpolates' between two specific formulations of the coupled flow (58) (cf. [57]).

In Eq. (59a), the differential $d_2 D_{\mathcal{F}}(f_i, f_j^*)$ means $dD_{\mathcal{F}}(f_i, \cdot)|_{f_j^*(t)}$ which determines the evolution $f_j^*(t)$ by averaging geometrically data points $\mathcal{F} = \{f_i\}_{i \in \mathcal{I}}$, using weights $\nu_{j|i}\big(W(t), F^*(t)\big)$ due to (60) that represent the current assignments of data points f_i, $i \in \mathcal{I}$ to the labels $f_j^*(t)$, $j \in \mathcal{J}$. This dependency on $W(t)$ *regularizes* the evolution $F^*(t)$.

Conversely, the dependency of $W(t)$ on $F^*(t)$ due to the right-hand side of (58b) is *implicitly* given through the concrete formulation (59b) in terms of the time-variant distances

$$D_i(t) = \big(D_{\mathcal{F}}(f_i, f_1^*(t)), \ldots, D_{\mathcal{F}}(f_i, f_c^*(t))\big)^\top \tag{61}$$

that generalize the likelihood map (29) and in turn (59b), through the similarity map (34).

In applications, a large number c of labels (57) is chosen so as to obtain an 'overcomplete' initial dictionary $F^*(0)$ in a preprocessing step. This helps to remove the bias caused by imperfect clustering at this initial stage of the overall algorithm. The final *effective* number c of labels $F^*(T)$ is smaller, however, and mainly determined by the scale of the assignment flow (cf. Remark 4): The regularizing effect of the assignments $W(t)$ on the evolution of labels $F^*(t)$ causes many labels $f_j^*(t)$ to merge or to 'die out', which can be recognized by weights $\nu_{j|i}\big(W(t), F^*(t)\big)$ converging to 0. Extracting the effective labels from $F^*(T)$ determines $\mathcal{F}_*$.

The benefit of the *unsupervised* assignment flow (59) is that the remaining labels moved to positions $f_j^*(T) \in \mathcal{F}$ that are difficult to determine beforehand in *supervised* scenarios.

3.2 Self-Assignment Flow: Learning Labels from Data

This section addresses the fundamental problem: How to determine labels $\mathcal{F}_*$ directly from given data $\mathcal{F}_n$ without any prior information? The resulting *self-assignment flow* generalizes the unsupervised assignment flow of Section 3.1 that is based on an initial label set $F^*(0)$ and label adaption.

A naive approach would set $F^*(0) = \mathcal{F}_n$ and apply the unsupervised assignment flow. In applications this is infeasible because n generally is large. We overcome this issue by marginalization as follows.

Let $\mathcal{F}_n' = \{f_1', \ldots, f_n'\}$ denote a copy of the given data and consider them as initial labels by setting $F^*(0) = \mathcal{F}_n'$. We interpret

$$W_{i,j} = \Pr(j|i), \qquad W_i \in \mathcal{S},\ j \in \mathcal{J},\ i \in \mathcal{I} \tag{62}$$

as posterior probabilities of assigning label f_j' to datum f_i, as discussed in [3]. Adopting the uninformative prior $\Pr(i) = \frac{1}{|\mathcal{I}|}$, $i \in \mathcal{I}$ and Bayes rule, we compute

$$\Pr(i|j) = \frac{\Pr(j|i)\Pr(i)}{\sum_{k \in \mathcal{I}} \Pr(j|k)\Pr(k)} = \big(WC(W)^{-1}\big)_{i,j}, \quad C(W) = \mathrm{Diag}(W^\top \mathbb{1}). \tag{63}$$

Next we determine the *probabilities of self-assignments* $f_i \leftrightarrow f_k$ of data points by marginalizing over the labels (data copies $\mathcal{F}_n'$) to obtain the *self-assignment matrix*

$$A_{k,i}(W) = \sum_{j \in J} \Pr(k|j) \Pr(j|i) = \left(WC(W)^{-1}W^\top\right)_{k,i}. \tag{64}$$

Note that the initial labels are no longer involved. Rather, their evolution as *hidden* variables is *implicitly* determined by the evolving assignments $W(t)$ and (63).

Finally, we replace the data term $\langle D, W\rangle = \sum_{i \in \mathcal{I}} \langle D_i, W_i\rangle$ of supervised scenarios (cf. Remark 3) by

$$E(W) = \langle D, A(W)\rangle, \tag{65}$$

with $D_{i,k} = d_{\mathcal{F}}(f_i, f_j)$ and $A(W)$ given by (64). In other words, we replace the assignment matrix W by the self-assignment matrix $A(W)$ that is *parametrized* by the assignment matrix, in order to generalize the data term from supervised scenarios to the current completely unsupervised setting.

As a consequence, we substitute the Euclidean gradient $\partial_{W_i} E(W))$ for the distances vectors (28) on which the likelihood map (29) is based. These likelihood vectors in turn generalize the similarity map (34) and thus define the **self-assignment flow** (35).

The approach has attractive properties that enable interpretations from various viewpoints. We mention here only two of them and refer to [59] for further discussion and to the forthcoming report [60].

(1) The self-assignment matrix $A(W)$ (64) may be seen as a weighted *adjacency matrix* of $\mathcal{G}$ and, in view of its entries, as a *self-affinity* matrix with respect to given data f_i, $i \in \mathcal{I}$ supported by $\mathcal{G}$. $A(W)$ is parametrized by $W(t)$ and (64) shows that it evolves in the cone of completely positive matrices [8]. This reflects the combinatorial nature of label learning problem, exhibits relations to *nonnegative matrix factorization* [12] and via convex duality to *graph partitioning* [42].

(2) $A(W)$ is nonnegative, symmetric and doubly stochastic. Hence it may be seen as *transportation plan* corresponding to the *discrete optimal transport problem* [40] of minimizing the objective function (65). Taking into account the factorization (64) and the parametrization by $W(t)$, minimizing the objective (65) may be interpreted as transporting the uniform prior measure $\Pr(i) = \frac{1}{|\mathcal{I}|}$, $i \in \mathcal{I}$ to the support of data points f_i that implicitly define latent labels. In this way, by means of the solution $W(t)$ to the self-assignment flow, labels $\mathcal{F}_*$ directly emerge from given data $\mathcal{F}_n$.

4 Regularization Learning by Optimal Control

A key component of the assignment flow is the similarity map (34) that couples single-vertex flows (30) within neighborhoods $\mathcal{N}_i$, $i \in \mathcal{I}$. Based on the 'context' in terms of data observed within these neighborhoods, the similarity map discriminates

structure from noise that is removed by averaging. In this section we describe how the weights (33) that parametrize the similarity map can be estimated from data [23].

Our approach is based on an approximation of the assignment flow that is governed by an ODE defined on the tangent space $\mathcal{T}_0$ which linearly depends on the weights (Section 4.1). Using this representation, the learning problem is subdivided into two tasks (Section 4.2):

(i) *Optimal weights* are computed from ground truth data and corresponding labelings.
(ii) A *prediction map* is computed in order to extrapolate the relation between observed data and optimal weights to novel data.

4.1 Linear Assignment Flow

We consider the following approximation of the assignment flow (35), introduced by [56].

$$\dot{W} = R_W\Big(S(W_0) + dS_{W_0} R_{W_0} \log \frac{W}{W_0}\Big), \qquad W(0) = W_0 = \mathbb{1}_{\mathcal{W}}. \tag{66}$$

The 'working point' $W_0 \in \mathcal{W}$ can be arbitrary, in principle. Numerical experiments [56, Section 6.3.1] showed, however, that using the barycenter $W_0 = \mathbb{1}_{\mathcal{W}}$ suffices for our purposes.

Assuming that elements of the tangent space $V \in \mathcal{T}_0 \subset \mathbb{R}^{n\times c}$ are written as vectors by stacking row-wise the tangent vectors V_i, $i \in \mathcal{I}$, the Jacobian dS_{W_0} is given by the sparse block matrix

$$dS_{W_0} = \big(A_{i,k}(W_0)\big)_{i,k\in\mathcal{I}}, \qquad A_{i,k}(W_0) = \begin{cases} w_{i,k} R_{S_i(W_0)}\big(\frac{V_k}{W_{0,k}}\big), & \text{if } k \in \mathcal{N}_i, \\ 0, & \text{otherwise.} \end{cases} \tag{67}$$

We call the nonlinear ODE (66) *linear assignment flow* because it admits the parametrization [56, Prop. 4.2]

$$W(t) = \operatorname{Exp}_{W_0}\big(V(t)\big), \tag{68a}$$

$$\dot{V} = R_{W_0}\big(S(W_0) + dS_{W_0} V\big), \quad V(0) = 0. \tag{68b}$$

Eq. (68b) is a *linear* ODE. In addition, Eq. (67) shows that it *linearly* depends on the weight parameters (33), which is convenient for estimating optimal values of these parameters.

4.2 Parameter Estimation and Prediction

Let

$$\mathcal{P} = \{\Omega_i : i \in \mathcal{I}\} \tag{69}$$

denote the parameter space comprising all 'weight patches' Ω_i according to (33), one patch assigned to every vertex $i \in \mathcal{I}$ within the corresponding neighborhood $\mathcal{N}_i$. Note that $\mathcal{P}$ is a parameter *manifold*: The space containing all feasible weight values of each patch Ω_i has the same structure (ignoring the different dimensions) as $\mathcal{S}$ given by (11).

Parameter estimation is done by solving the constrained optimization problem

$$\min_{\Omega \in \mathcal{P}} \; E\big(V(T)\big) \tag{70a}$$

$$\text{s.t. } \dot{V} = f(V, \Omega), \qquad t \in [0, T], \qquad V(0) = 0, \tag{70b}$$

where (70b) denotes the linear ODE (68b) and the essential variables V and $\Omega = \{\Omega_1, \Omega_2, \ldots, \Omega_n\}$ (all weight patches) in compact form. A basic instance of the objective function (70a) is

$$E\big(V(T)\big) = D_{\mathrm{KL}}\big(W^*, \mathrm{Exp}_{W_0}\big(V(t)\big)\big), \tag{71}$$

which evaluates the Kullback-Leibler divergence of the labeling induced by $V(T)$ by (68a) from a given ground-truth labeling $W^* \in \mathcal{W}$.

Problem (70) can be tackled in two ways as indicated by the following diagram.

$$\begin{array}{ccc} E\big(V(T)\big) \text{ s.t. } \dot{V} = f(V,\Omega) & \xrightarrow{\text{differentiate}} & \textbf{adjoint system} \\ \Big\downarrow{\scriptstyle \text{discretize}} & & \Big\downarrow{\scriptstyle \text{discretize}} \\ \textbf{nonlinear program} & \xrightarrow[\text{differentiate}]{} & \textbf{sensitivity} \end{array} \tag{72}$$

Differentiation yields the adjoint system which, together with the primal system (70b) and proper discretization, enables to compute the sensitivity $\frac{\mathrm{d}}{\mathrm{d}\Omega} E\big(V(T)\big)$ by numerical integration. Alternatively, one first selects a numerical scheme for integrating the primal system (70b) which turns (70) into a nonlinear program that can be tackled by established methods.

Most appealing are situations where these two approaches are equivalent, that is when the above diagram commutes [44]. A key aspect in this context concerns the symplectic numerical integration of the joint system. We refer to [23] for details and to [48] for the general background.

The weight parameters are updated by numerically integrating the Riemannian gradient descent flow

$$\dot{\Omega} = -\operatorname{grad}_{\mathcal{P}} E\big(V(T)\big) = -R_\Omega \frac{\mathrm{d}}{\mathrm{d}\Omega} E\big(V(T)\big), \quad \Omega(0) = \mathbb{1}_{\mathcal{P}}, \tag{73}$$

based on the sensitivities determined using either (equivalent) path of diagram (72). The linear map R_Ω factorizes according to (69) into components R_{Ω_i}, $i \in \mathcal{I}$ that are given by (18) and well-defined due to (33).

Running this algorithm for many instances of data $\mathcal{F}_n^1, \mathcal{F}_n^2, \dots$ and corresponding ground-truth labelings $W^{*1}, W^{*2}, \dots$ produces the optimal weights $\Omega^{*1}, \Omega^{*2}, \dots,$

$$\big\{\{\mathcal{F}_n^1, \mathcal{F}_n^2, \dots\}, \{W^{*1}, W^{*2}, \dots\}\big\} \quad \longrightarrow \quad \big\{\{\mathcal{F}_n^1, \mathcal{F}_n^2, \dots\}, \{\Omega^{*1}, \Omega^{*2}, \dots\}\big\}. \tag{74}$$

We rearrange the data *patch-wise* and denote them by $\mathcal{F}_1^*, \mathcal{F}_2^*, \dots,$ i.e. $\mathcal{F}_i^*$ denotes a feature patch[4] extracted in *any* order from some $\mathcal{F}_n^k$. Grouping these feature patches with the corresponding optimal weight patches, extracted from $\Omega^{*1}, \Omega^{*2}, \dots$ in the *same* order, yields the input data

$$\big\{(\mathcal{F}_1^*, \Omega_1^*), \dots, (\mathcal{F}_N^*, \Omega_N^*)\big\} \tag{75}$$

for prediction, possible after data size reduction by condensing it to a coreset [41]. The predictor

$$\widehat{\omega} \colon \mathcal{F} \to \mathcal{P}, \qquad \mathcal{F}_i \mapsto \Omega_i \tag{76}$$

returns for any feature patch $\mathcal{F}_i \subset \mathcal{F}_n$ of *novel* data $\mathcal{F}_n$ a corresponding weight patch Ω_i (33) that controls the similarity map (34).

A basic example of a predictor map (76) is the *Nadaraya-Watson* kernel regression estimator [52, Section 5.4]

$$\widehat{\omega}(\mathcal{F}_i) = \sum_{k \in [N]} \frac{K_h(\mathcal{F}_i, \mathcal{F}_k^*)}{\sum_{k' \in [N]} K_h(\mathcal{F}_i, \mathcal{F}_{k'}^*)} \Omega_k^* \tag{77}$$

with a proper kernel function (Gaussian, Epanechnikov, etc.) and bandwidth parameter estimated, e.g., by cross-validation based on (75). We refer to [23] for numerical examples.

Remark 5 (Feasibility of learning.)

The present notion of *context* is quite limited: it merely concerns the co-occurrence of features within local neighborhoods $\mathcal{N}_i$. This limits the scope of the assignment flow for applications, so far.

On the other hand, this limited scope enables to subdivide the problem of *learning* these contextual relationships into two *manageable tasks* (i), (ii) mentioned in the first paragraph of this section: Subtask (i) can be solved using sound numerics (recall the discussion of (72)) without the need to resort to opaque toolboxes, as is common in machine learning. Subtask (ii) can be solved using a range of state-of-the-art methods of computational statistics and machine learning, respectively.

The corresponding situation seems less clear for more complex networks that are empirically investigated in the current literature on machine learning. Therefore, the strategy to focus first on the relations between data, data structure and label

[4] Not to be confused with *labels* $\mathcal{F}_*$!

assignments at *two adjacent scales* (vertices $\leftrightarrow$ neighborhoods $\mathcal{N}_i \leftrightarrow$ neighborhoods of neighborhoods, and so forth) appears to be more effective, in the long run.

5 Outlook

This project has started about two years ago. Motivation arises from computational difficulties encountered with the evaluation of hierarchical discrete graphical models and from our limited mathematical understanding of deep networks. We outline our next steps and briefly comment on a long-term perspective.

Current Work. Regarding *unsupervised learning*, we are focusing on the low-rank structure of the factorized self-assignment matrix (64) that is caused by the *regularization* of the assignment flow and corresponds to the reduction of the effective number of labels (cf. the paragraph below Eq. (61)). Our objective is to learn labels directly from data in terms of *patches of assignments* for any class of images at hand.

It is then a natural consequence to extend the objective (71) of *controlling* the assignment flow to such dictionaries of assignment patches, that encode image structure at the subsequent local scale (measured by $|\mathcal{N}_i|$). In addition, the prediction map (77) should be generalized to *feedback* control that not only takes into account feature similarities, but also similarities between the current state $W(t)$ of the assignment flow and assignment trajectories $W^{*k}(t)$. The latter are computed anyway when estimating the parameters on the right-hand side of (74) from the data on the left-hand side.

Coordinating in this way unsupervised learning and control using the assignment flow will satisfactorily solve our current core problem discussed as Remark 5.

Perspective. In order to get rid of discretization parameters, we are currently studying variants of the assignment flow on continuous domains [50]. 'Continuous' here not only refers to the underlying Euclidean domain Ω replacing the graph $\mathcal{G}$, but also to the current *discrete* change of scale $i \to |\mathcal{N}_i|$, that should become infinitesimal and *continuous*. This includes a continuous-domain extension of the approach [49], where a variational formulation of the assignment flow was studied that is inline with the *additive* combination of data term and regularization in related work [53, 7]. Variational methods (Γ-convergence, harmonic maps) then may provide additional mathematical insight into the regularization property of the assignment flow, into a geometric characterization of partitions of the underlying domain, and into the pros and cons of the compositional structure of the assignment flow.

Acknowledgements I thank my students Ruben Hühnerbein, Fabrizio Savarino, Alexander Zeilmann, Artjom Zern, Matthias Zisler and my colleague Stefania Petra for many discussions and collaboration. We gratefully acknowledge support by the German Science Foundation (DFG), grant GRK 1653.

This work has also been stimulated by the recently established Heidelberg STRUCTURES Excellence Cluster, funded by the DFG under Germany's Excellence Strategy EXC-2181/1 - 390900948.

References

1. Amari, S.I., Nagaoka, H.: Methods of Information Geometry. Amer. Math. Soc. and Oxford Univ. Press (2000)
2. Antun, V., Renna, F., Poon, C., Adcock, B., Hansen, A.C.: On Instabilities of Deep Learning in Image Reconstruction - Does AI Come at a Cost? CoRR abs/1902.05300 (2019)
3. Åström, F., Petra, S., Schmitzer, B., Schnörr, C.: Image Labeling by Assignment. Journal of Mathematical Imaging and Vision **58**(2), 211–238 (2017)
4. Ay, N., Jost, J., Lê, H.V., Schwachhöfer, L.: Information Geometry. Springer (2017)
5. Barndorff-Nielsen, O.E.: Information and Exponential Families in Statistical Theory. Wiley, Chichester (1978)
6. Basseville, M.: Divergence Measures for Statistical Data Processing – An Annotated Bibliography. Signal Proc. **93**(4), 621–633 (2013)
7. Bergmann, R., Tenbrinck, D.: A Graph Framework for Manifold-Valued Data. SIAM Journal on Imaging Sciences **11**(1), 325–360 (2018)
8. Berman, A., Shaked-Monderer, N.: Completely Positive Matrices. World Sci. Publ. (2003)
9. Boykov, Y., Veksler, O., Zabih, R.: Fast Approximate Energy Minimization via Graph Cuts. IEEE Trans. Patt. Anal. Mach. Intell. **23**(11), 1222–1239 (2001)
10. Calin, O., Udriste, C.: Geometric Modeling in Probability and Statistics. Springer (2014)
11. Chan, T.F., Esedoglu, S., Nikolova, M.: Algorithms for Finding Global Minimizers of Image Segmentation and Denoising Models. SIAM J. Appl. Math. **66**(5), 1632–1648 (2006)
12. Cichocki, A., Zdunek, A., Phan, A.H., Amari, S.I.: Nonnegative Matrix and Tensor Factorizations. John Wiley & Sons (2009)
13. Cortes, C., Vapnik, V.: Support-Vector Networks. Mach. Learning **20**, 273–297 (1995)
14. Cover, T.M., Thomas, J.A.: Elements of Information Theory, 2nd edn. John Wiley & Sons (2006)
15. Elad, M.: Deep, Deep Trouble: Deep Learning's Impact on Image Processing, Mathematics, and Humanity. SIAM News (2017)
16. Gary, R.M., Neuhoff, D.L.: Quantization. IEEE Trans. Inform. Theory **44**(6), 2325–2383 (1998)
17. Geman, S., Geman, D.: Stochastic Relaxation, Gibbs Distributions, and the Bayesian Restoration of Images. IEEE Trans. Patt. Anal. Mach. Intell. **6**(6), 721–741 (1984)
18. Graf, S., Luschgy, H.: Foundations of Quantization for Probability Distributions, *Lect. Notes Math.*, vol. 1730. Springer (2000)
19. Hairer, E., Lubich, C., Wanner, G.: Geometric Numerical Integration. Springer (2006)
20. Har-Peled, S.: Geometric Approximation Algorithms. AMS (2011)
21. Hofbauer, J., Siegmund, K.: Evolutionary Game Dynamics. Bull. Amer. Math. Soc. **40**(4), 479–519 (2003)
22. Hühnerbein, R., Savarino, F., Åström, F., Schnörr, C.: Image Labeling Based on Graphical Models Using Wasserstein Messages and Geometric Assignment. SIAM J. Imaging Science **11**(2), 1317–1362 (2018)
23. Hühnerbein, R., Savarino, F., Petra, S., Schnörr, C.: Learning Adaptive Regularization for Image Labeling Using Geometric Assignment. In: Proc. SSVM. Springer (2019)
24. Hummel, R.A., Zucker, S.W.: On the Foundations of the Relaxation Labeling Processes. IEEE Trans. Patt. Anal. Mach. Intell. **5**(3), 267–287 (1983)
25. Idel, M.: A Review of Matrix Scaling and Sinkhorn's Normal Form for Matrices and Positive Maps. CoRR abs/1609.06349 (2016)
26. Iserles, A., Munthe-Kaas, H.Z., Nørsett, S.P., Zanna, A.: Lie-group methods. Acta Numerica **14**, 1–148 (2005)
27. Jost, J.: Riemannian Geometry and Geometric Analysis, 7th edn. Springer-Verlag Berlin Heidelberg (2017)
28. 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. Computer Vision **115**(2), 155–184 (2015)

29. Kleinberg, J., Tardos, E.: Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. J. ACM **49**(5), 616–639 (2002)
30. Koller, D., Friedman, N.: Probabilistic Graphical Models: Principles and Techniques. MIT Press (2009)
31. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet Classification with Deep Convolutional Neural Networks. In: Adv. NIPS, pp. 1097–1105 (2012)
32. Lauritzen, S.L.: Chapter 4: Statistical Manifolds. In: S.S. Gupta, S.I. Amari, O.E. Barndorff-Nielsen, R.E. Kass, S.L. Lauritzen, C.R. Rao (eds.) Differential Geometry in Statistical Inference, pp. 163–216. Institute of Mathematical Statistics, Hayward, CA (1987)
33. Lauritzen, S.L.: Graphical Models. Clarendon Press, Oxford (1996)
34. Lee, J.M.: Introduction to Smooth Manifolds. Springer (2013)
35. Lellmann, J., Schnörr, C.: Continuous Multiclass Labeling Approaches and Algorithms. SIAM J. Imag. Sci. **4**(4), 1049–1096 (2011)
36. Mézard, M., Montanari, A.: Information, Physics, and Computation. Oxford Univ. Press (2009)
37. Munthe-Kaas, H.: High Order Runge-Kutta Methods on Manifolds. Applied Numerical Mathematics **29**(1), 115–127 (1999)
38. Pavan, M., Pelillo, M.: Dominant Sets and Pairwise Clustering. IEEE Trans. Patt. Anal. Mach. Intell. **29**(1), 167–172 (2007)
39. Pelillo, M.: The Dynamics of Nonlinear Relaxation Labeling Processes. J. Math. Imag. Vision **7**, 309–323 (1997)
40. Peyré G.and Cuturi, M.: Computational Optimal Transport. CNRS (2018)
41. Phillips, J.: Coresets and Sketches. In: Handbook of Discrete and Computational Geometry, chap. 48. CRC Press (2016)
42. Povh, J., Rendl, F.: A Copositive Programming Approach to Graph Partitioning. SIAM J. Optimization **18**(1), 223–241 (2007)
43. Rosenfeld, A., Hummel, R.A., Zucker, S.W.: Scene Labeling by Relaxation Operations. IEEE Trans. Systems, Man, and Cyb. **6**, 420–433 (1976)
44. Ross, I.: A Roadmap for Optimal Control: The Right Way to Commute. Annals of the New York Academy of Sciences **1065**(1), 210–231 (2006)
45. Rudin, L., Osher, S., Fatemi, E.: Nonlinear Total Variation Based Noise Removal Algorithms. Physica D **60**, 259–268 (1992)
46. Rumelhart, D.E., McClelland, J.L.: Parallel Distributed Processing: Explorations in the Microstructure of Cognition: Foundations. MIT Press (1986)
47. Sandholm, W.H.: Population Games and Evolutionary Dynamics. MIT Press (2010)
48. Sanz-Serna, J.: Symplectic Runge–Kutta Schemes for Adjoint Equations, Automatic Differentiation, Optimal Control, and More. SIAM Review **58**(1), 3–33 (2016)
49. Savarino, F., Schnörr, C.: A Variational Perspective on the Assignment Flow. In: Proc. SSVM. Springer (2019)
50. Savarino, F., Schnörr, C.: Assignment Flows on Continuous Domains: Likelihood Flows in Harmony. Tech. rep., Heidelberg University (2019, in preparation)
51. Wainwright, M.J., Jordan, M.I.: Graphical Models, Exponential Families, and Variational Inference. Found. Trends Mach. Learn. **1**(1-2), 1–305 (2008)
52. Wasserman, L.: All of Nonparametric Statistics. Springer (2006)
53. Weinmann, A., Demaret, L., Storath, M.: Total Variation Regularization for Manifold-Valued Data. SIAM J. Imag. Sci. **7**(4), 2226–2257 (2014)
54. Werner, T.: A Linear Programming Approach to Max-sum Problem: A Review. IEEE Trans. Patt. Anal. Mach. Intell. **29**(7), 1165–1179 (2007)
55. Yedidia, J.S., Freeman, W.T., Weiss, Y.: Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms. IEEE Trans. Information Theory **51**(7), 2282–2312 (2005)
56. Zeilmann, A., Savarino, F., Petra, S., Schnörr, C.: Geometric Numerical Integration of the Assignment Flow. Inverse Problems, https://doi.org/10.1088/1361-6420/ab2772, in press (2019)
57. Zern, A., Zisler, M., Petra, S., Schnörr, C.: Unsupervised Assignment Flow: Label Learning on Feature Manifolds by Spatially Regularized Geometric Assignment. CoRR abs/1904.10863 (2019)

58. Zern, A., Zisler, M., Åström, F., Petra, S., Schnörr, C.: Unsupervised Label Learning on Manifolds by Spatially Regularized Geometric Assignment. In: Proc. GCPR. Springer (2018)
59. Zisler, M., Zern, A., Petra, S., Schnörr, C.: Unsupervised Labeling by Geometric and Spatially Regularized Self-Assignment. In: Proc. SSVM. Springer (2019)
60. Zisler, M., Zern, A., Petra, S., Schnörr, C.: Self-Assignment Flows for Data Labeling and Unsupervised Label Learning. Tech. rep., Heidelberg University (2019, in preparation)