UNSUPERVISED ASSIGNMENT FLOW: LABEL LEARNING ON FEATURE MANIFOLDS BY SPATIALLY REGULARIZED GEOMETRIC ASSIGNMENT

ARTJOM ZERN, MATTHIAS ZISLER, STEFANIA PETRA, CHRISTOPH SCHNÖRR

ABSTRACT. This paper introduces the *unsupervised assignment flow* that couples the assignment flow for supervised image labeling [ÅPSS17] with Riemannian gradient flows for label evolution on feature manifolds. The latter component of the approach encompasses extensions of state-of-the-art clustering approaches to manifold-valued data. Coupling label evolution with the spatially regularized assignment flow induces a sparsifying effect that enables to learn compact label dictionaries in an unsupervised manner. Our approach alleviates the requirement for supervised labeling to have proper labels at hand, because an initial set of labels can evolve and adapt to better values while being assigned to given data. The separation between feature and assignment manifolds enables the flexible application which is demonstrated for three scenarios with manifold-valued features. Experiments demonstrate beneficial effect in both directions: adaptivity of labels improves image labeling, and steering label evolution by spatially regularized assignments leads to proper labels, because the assignment flow for supervised labeling is exactly used without any approximation for label learning.

CONTENTS

Key words and phrases. assignment flow, assignment manifold, divergence function, Stein divergence, feature manifolds, positive definite matrices, covariance descriptors, unsupervised learning, clustering, information geometry.

This work is supported by Deutsche Forschungsgemeinschaft (DFG) under Germanys Excellence Strategy EXC-2181/1 - 390900948 (the Heidelberg STRUCTURES Excellence Cluster) and by the Research Training Group funded by the DFG, grant GRK 1653.

1. Introduction

1.1. **Motivation.** Geometric methods based on manifold models of data and Riemannian geometry are nowadays widely employed in image processing and computer vision [TS16]. For example, covariance descriptors play a prominent role [TS16, CS16]. Covariance descriptors are typically applied to the detection and classification of entire images (e.g. faces, texture) or videos (e.g. action recognition), or as desciptors of local image structure. An important task in this context is to compute a *codebook* of covariance descriptors that can be used for solving a task at hand like, e.g., image classification by nearest-neighbor search [CSBP13], or image labeling [KAH+15] using the codebook descriptors as labels.

The recent work [HHLS16] introduced a state-of-the-art method for computing such codebooks. After embedding descriptors into a reproducing kernel Hilbert space [HSS08], given data are approximated by a kernel expansion, and a sparse subset can be determined by ℓ_1-regularization of the expansion coefficients. This method works *entirely in feature space*, however, and ignores the *spatial structure of codebook assignments* to data, which is unfavorable in connection with image labeling. Figure 1.1 illustrates why the spatial structure of label assignments should also drive the *evolution of labels* in feature space for *unsupervised* label learning, if the resulting labels are subsequently used for *supervised* image labeling for which spatial regularization is typically enforced as well.

We show in this paper how the approach of [ÅPSS17] to spatially regularized label assignment can be combined with basic clustering approaches after extending the latter to feature manifolds, to perform *unsupervised label learning* from manifold-valued feature data through spatially regularized label assignment. Our approach is *consistent and natural* in that the *very same* approach [ÅPSS17] for supervised image labeling is also used for the unsupervised learning of proper labels for this task.

1.2. **Related Work, Contribution.** The classical approach for the unsupervised learning of feature prototypes ('labels') is the mean-shift iteration [FH75, CM02], which iteratively seeks modes (local peaks) of the feature density distribution through the averaging of features within local neighborhoods. This has been generalized to *manifold-valued* features by [SM09], by replacing ordinary mean-shifts by Riemannian (Fréchet, Karcher) means [Kar77]. The common way to take into account the *spatial structure* of label assignments is to *augment* the feature space by *spatial coordinates*, e.g. by turning a color feature (r, g, b) into the feature vector (x, y, r, g, b). This *merge* of feature space and spatial domain has a conceptual drawback, however: The *same* color vector $(\overline{r}, \overline{g}, \overline{b})$ observed at two *different* locations $(x_1, y_1, \overline{r}, \overline{g}, \overline{b})$, $(x_2, y_2, \overline{r}, \overline{g}, \overline{b})$ defines

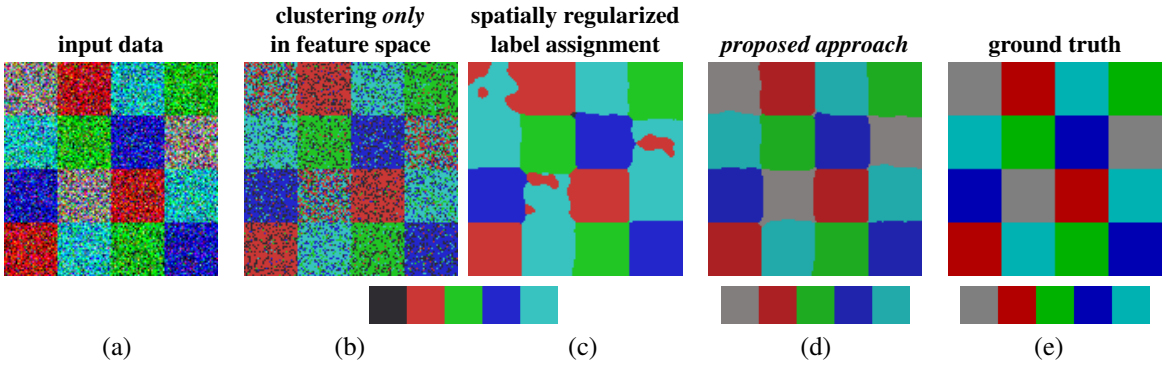

FIGURE 1.1. **Importance of spatially regularized assignments for label *learning*.** (a) Input data: a synthetic image corrupted by Gaussian noise. (b) + (c) The classical two-step approach of clustering in feature space first (panel b) followed by *supervised* label assignment (panel c) performs poorly, despite spatial regularization. (d) By *coupling* label evolution and spatially regularized assignment both the label set and the labeled image can be drastically improved. (e) Ground truth labeling and label set. Label set resulting from (b), (d) and (e) are depicted below the respective image labeling results.

two *different* feature vectors, and hence these two feature vectors may be assigned to different prototypes during clustering despite containing the same color information. Furthermore, clustering spatial coordinates into *centroids* by mean-shifts (together with the features) *differs* from *unbiased spatial* regularization as performed by variational approaches, graphical models or the assignment flow approach of [ÅPSS17], where regularization does *not* depend on the location of centroids and the corresponding shape of local density modes.

We introduce a novel approach which has the following properties:

(i) The approach incorporates and performs *unsupervised learning of manifold-valued features*, henceforth called *labels*. The approach applies to any feature manifold [SM09] for which the corresponding operations defined below like, e.g., Riemannian means are well-defined and computationally feasible. Experiments using S^1-valued data (2D-orientations), $\mathrm{SO}(3)$-valued data (orthogonal frames) and features on the positive definite matrix manifold (covariance descriptors) illustrate our approach.

(ii) The evolution of labels (unsupervised learning) is driven by *spatially regularized assignments* which are *not* biased towards any spatial centroids. This is accomplished by applying the smooth geometric assignment approach to image labeling recently introduced by [ÅPSS17].

(iii) The smooth settings of both (i), (ii) enable to define a *smooth coupled flow*

$$(\dot{G}, \dot{W}) = \mathcal{V}(G, W) \tag{1.1}$$

where $\dot{G}$ denotes the evolution of labels and $\dot{W}$ the evolution of spatially regularized label assignments that *interact* through a coupling vector field $\mathcal{V}$. This interaction keeps both domains (i) and (ii) *separate* and hence enables to apply flexibly our approach to various feature manifolds, using the *same* regularized assignment mechanism.

A preliminary version of this paper [ZZr+18] introduced the approach called 'coupled flow A' in this paper. The present paper elaborates this conference paper in many ways as illustrated by Fig. 1.2, including a more comprehensive experimental evaluation. In particular, we provide a more general natural definition of a one-parameter family of *unsupervised assignment flows* that smoothly interpolate 'coupled flow A' and the novel, more general 'coupled flow B', including these two flows as special cases.

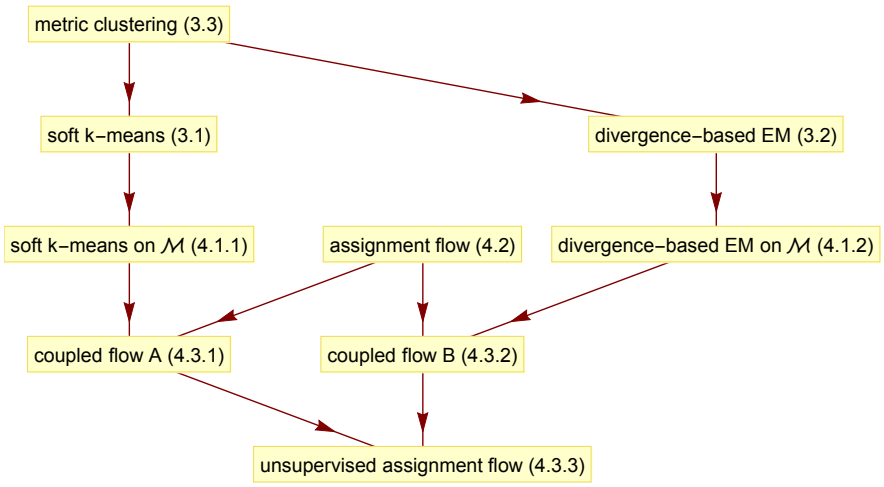

FIGURE 1.2. **Organization of this paper** (with respective section numbers in brackets): *metric clustering* provides an overcomplete set of labels in a preprocessing step, based on which both *soft k-means* and the *EM-iteration*, defined by a divergence function for modelling class-conditional distributions, perform classical label evolution by mean-shift iterations. The latter two approaches are generalized to feature data taking values in a Riemannian manifold $\mathcal{M}$, and coupled with the assignment flow that induces a sparsifying effect through spatial regularization. We show that the resulting *coupled flow A* is a special case of the *coupled flow B*. A corresponding interpolating flow finally defines the *unsupervised assignment flow*.

1.3. **Basic notation.** We set $[n] = \{1, 2, \ldots, n\}$ for $n \in \mathbb{N}$ and $\mathbb{1} = (1, 1, \ldots, 1)^\top$ with dimension depending on the context. Euclidean vectors are enumerated by superscripts with components indexed by subscripts $x^i = (x^i_1, \ldots, x^i_d)^\top \in \mathbb{R}^d$. $\langle \cdot, \cdot \rangle$ denotes the Euclidean inner product $\langle u, v \rangle = \sum_{i \in [d]} u_i v_i$ of two vectors or the Frobenius inner product $\langle A, B \rangle = \operatorname{tr}(A^\top B)$ for matrices. Throughout the paper, the symbols I, J denote

$$\begin{aligned} I &: \text{ set of data indices}, \\ J &: \text{ labels of label indices}, \end{aligned} \tag{1.2}$$

with cardinalities $|I|$ and $|J|$. The relation $A \succ 0$ ($A \succeq 0$) indicates that a symmetric matrix $A = A^\top$ is positive (semi-) definite. The $(d-1)$-dimensional probability simplex is denoted by

$$\Delta_d = \big\{ x \in \mathbb{R}^d \colon x_i \geq 0,\ i \in [d],\ \langle \mathbb{1}, x \rangle = 1 \big\} \subset \mathbb{R}^d. \tag{1.3}$$

For *strictly* positive probability vectors $0 < p, q \in \Delta_n$, we denote *componentwise* multiplication and subdivision efficiently by pq and $\frac{p}{q}$, respectively. It will be convenient to denote the exponential function with vectors as argument in two alternative ways,

$$\exp(x) = e^x = (e^{x_1}, \ldots, e^{x_d})^\top, \tag{1.4}$$

and similarly with $\log(x)$ and strictly positive vectors $x > 0$. $(\mathcal{M}, g)$ *generally* denote some Riemannian manifold $\mathcal{M}$ with metric g, whereas $\mathcal{S}, \mathcal{W}, \mathcal{P}_d$ denote *specific* Riemannian manifolds defined in subsequent sections. In this context the symbol $\exp_p$ *with* subscript denotes the exponential map of $\mathcal{M}$ and should not be confused with the exponential function (1.4) that is uniquely denoted *without* subscript.

2. BACKGROUND

This section collects background material required in this paper. Section 2.1 covers basic notion of differential geometry. We recommend [Lee13, Jos17] for further reading. Section 2.2 recalls the notion of a *divergence function*. Such functions are used in applications in lieu of the squared Riemannian distance if evaluating the latter is computationally too involved. See [Bas13] for a survey and [AC10] for a mathematical account. Concrete divergence functions will be considered in Section 5.

2.1. Basic Notions from Differential Geometry.

Let $(\mathcal{M}, g)$ be a Riemannian manifold with metric g. We denote by $\mathcal{F}(\mathcal{M})$ the set of smooth functions $f\colon \mathcal{M} \to \mathbb{R}$ and by $\mathfrak{X}(\mathcal{M})$ and $\mathfrak{X}^*(\mathcal{M})$ the set of all smooth vector and covector fields on $\mathcal{M}$. Tangent and cotangent spaces are denoted by $T_p\mathcal{M}$ and $T_p^*\mathcal{M}$, $p \in \mathcal{M}$ and subscripts $X_p \in T_p\mathcal{M}$ indicate the evaluation of a vector field $X \in \mathfrak{X}(\mathcal{M})$. $df \in \mathfrak{X}^*(\mathcal{M})$ denotes the differential of a function $f \in \mathcal{F}(\mathcal{M})$ and $df(X)$ and $df_p(v)$ its action on $X \in \mathfrak{X}(\mathcal{M})$ and $v \in T_p\mathcal{M}$. We use both notations

$$g(X,Y) = \langle X, Y\rangle_g, \quad X, Y \in \mathfrak{X}(\mathcal{M}) \tag{2.1}$$

when evaluating the metric. The *Riemannian gradient* of a function $f \in \mathcal{F}(\mathcal{M})$ is the vector field

$$\operatorname{grad} f \in \mathfrak{X}(\mathcal{M}) \tag{2.2a}$$

defined by

$$\langle \operatorname{grad} f, X\rangle_g = df(X) = Xf, \qquad \forall X \in \mathfrak{X}(\mathcal{M}). \tag{2.2b}$$

Let $\widehat{g}$ denote the linear tangent-cotangent isomorphism

$$\widehat{g}\colon \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}^*(\mathcal{M}), \qquad \widehat{g}(X)(Y) = g(X,Y), \quad \forall X, Y \in \mathfrak{X}(\mathcal{M}) \tag{2.3}$$

that associates with a vector field X the covector field $\widehat{g}(X) = g(X, \cdot)$. Then by (2.2b),

$$\operatorname{grad} f = \widehat{g}^{-1}(df) \tag{2.4}$$

The *exponential map* at p

$$\exp_p\colon V_p \to \mathcal{M}, \qquad v \mapsto \exp_p(v) = \gamma_v(1) \tag{2.5a}$$

is defined on

$$V_p = \{v \in T_p\mathcal{M}\colon \gamma_v \text{ is defined on } [0,1]\} \tag{2.5b}$$

in terms of the *geodesic* $\gamma_v(t)$ through $p = \gamma_v(0)$ with velocity $v = \dot\gamma_v(0)$.

The *weighted Riemannian mean* [Jos17, Def. 6.9.1] of a collection $p_1, \dots, p_n \in \mathcal{M}$ of points with respect to weights $w = (w_1, \dots, w_n) \in \Delta_n$ is the point $q \in \mathcal{M}$ satisfying

$$J_w(q) = \inf_{p \in \mathcal{M}} J_w(p), \qquad J_w(p) = \frac{1}{2}\sum_{i\in[n]} w_i d_g^2(p_i, p), \tag{2.6}$$

where $d_g(q,p)$ denotes the *Riemannian distance*, i.e. the infimum of the length of all smooth paths connecting q and p on $\mathcal{M}$. We have [Jos17, Lemma 6.9.4]

$$\operatorname{grad}_p J_w = -\sum_{i\in[n]} w_i \exp_p^{-1}(p_i) \in T_p\mathcal{M} \tag{2.7a}$$

and hence the optimality condition for q

$$\sum_{i\in[n]} w_i \exp_q^{-1}(p_i) = 0. \tag{2.7b}$$

This equation is typically solved by the *mean shift* (fixed point) *iteration*

$$q^{(t+1)} = \exp_{q^{(t)}}\Big(\sum_{i\in[n]} w_i \exp^{-1}_{q^{(t)}}(p_i)\Big), \quad t = 1, 2, \dots \tag{2.8}$$

with a suitable initialization $q^{(0)}$.

2.2. **Divergence Functions.** *Bregman divergences* are distance-like functions of the form

$$D_\phi \colon \operatorname{dom}\phi \times \operatorname{int}(\operatorname{dom}\phi) \to \mathbb{R}_+, \qquad D_\phi(x, y) = \phi(x) - \phi(y) - \langle \nabla\phi(y), x - y\rangle, \tag{2.9}$$

induced by smooth convex functions $\phi \colon \mathbb{R}^d \to \mathbb{R}$ of Legendre type [CZ97, BB97]. Divergences D_ϕ satisfy

$$D_\phi \geq 0 \quad \text{and} \quad D_\phi(x, y) = 0 \;\Leftrightarrow\; x = y, \tag{2.10a}$$

$$\nabla^2_x D_\phi(x, y) \succ 0, \quad \forall x \in \mathbb{R}^d. \tag{2.10b}$$

The former property shows that D_ϕ behave like a distance, but symmetry $D_\phi(x, y) = D_\phi(y, x)$ is not required and generally does not hold. The second property shows that D_ϕ can be used to define a metric in order to turn an open subset of a Euclidean space into a manifold.

More generally, given a d-dimensional Riemannian manifold $(\mathcal{M}, g)$, a function $D_\phi \colon \mathcal{M} \times \mathcal{M} \to \mathbb{R}_+$ is a proper divergence function defined on $\mathcal{M}$ if, for any chart $U \subset \mathcal{M}$ with local coordinates $x \colon U \to \mathbb{R}^d$ and $p, q \in U$, the function

$$\widetilde{D}_\phi(x, y) = \widetilde{D}_\phi\big(x(p), x(q)\big) = D_\phi(p, q) \tag{2.11a}$$

satisfies (2.10a) and recovers the positive definite metric tensor by

$$D_\phi(p, q) \approx \frac{1}{2} \sum_{i,j\in[d]} g_{ij}(p) z_i z_j, \tag{2.11b}$$

for $z = x(q) - x(p)$ and small $\|z\|$.

Two further facts are relevant for the present paper. Firstly, the function $D_\phi(p, q) = \frac{1}{2} d_g(p, q)^2$ defines a *canonical* divergence function on a Riemannian manifold $(\mathcal{M}, g)$ in terms of the squared Riemannian distance d^2_g. Secondly, alternative divergence functions D_ϕ satisfying (2.11) are required in many applications, that serve as surrogate functions in (2.6) for the squared Riemannian distance $d^2_g(p_i, p)$ and are easier to evaluate computationally. Concrete divergence functions in connection with unsupervised label learning will be studied in Section 5.

Likewise, in information geometry, the Riemannian (Levi-Civita) connection is replaced by another affine connection in order to define a divergence function through affine geodesics and corresponding squared distances. We refer to [AN00, section 3.4], [AC10] and [AJLS17, section 4.4] for background and further details. A concrete application are provided by the assignment manifold (Section 4.2) and corresponding concepts defining the assignment flow in Section 4.2.

3. Basic Clustering

We briefly summarize in this section the basic iterative schemes

- soft-k-means clustering in Euclidean spaces (Section 3.1),
- clustering using mixture distributions, divergence functions and the EM-algorithm (Section 3.2), and
- greedy-based clustering in metric spaces (Section 3.3).

The former two approaches will be generalized to *manifold-valued* data (features) in Section 4.1 and coupled with the assignment flow for spatial regularization in Section 4.3.

Metric clustering applies to any metric space, in particular to manifolds with the Riemannian distance or a suitable divergence as surrogate distance function. The method has linear complexity and comes along

with a performance guarantee. Hence this method is suited for fast data selection in a preprocessing step, to obtain an overcomplete codebook (set of prototypes) as initialization for manifold-valued clustering, which subsequently optimizes and sparsifies this codebook in a computationally more expensive way.

3.1. **Euclidean Soft-k-Means Clustering.** The content of this paragraph can be found in numerous papers and textbooks. We merely refer to the survey [Teb07] and to the bibliography therein.

Given data vectors $x^1, \dots, x^{|I|} \in \mathbb{R}^d$, we consider the task of determining prototype vectors

$$M = \{m^1, \dots, m^{|J|}\} \subset \mathbb{R}^d \tag{3.1}$$

by minimizing the k-means criterion[1]

$$E(M) = \sum_{i \in I} \min_{j \in J} \|x^i - m^j\|^2 = \sum_{i \in I} \operatorname{vecmin}\big(D_i(M)\big), \tag{3.2}$$

where

$$D_i(M) = \big(D_{i1}(M), \dots, D_{i|J|}(M)\big) = \big(\|x^i - m^1\|^2, \dots, \|x^i - m^{|J|}\|^2\big) \tag{3.3}$$

and

$$\operatorname{vecmin}(z) = \min_{j \in [d]} \{z_1, \dots, z_d\}, \quad z \in \mathbb{R}^d, \quad d \in \mathbb{N}. \tag{3.4}$$

Soft-k-means is based on the *smoothed* objective

$$E_\varepsilon(M) = -\varepsilon \sum_{i \in I} \log \Big(\sum_{j \in J} \exp\big(- \frac{\|x^i - m^j\|^2}{\varepsilon} \big) \Big), \quad \varepsilon > 0 \tag{3.5}$$

which results from approximating the inner minimization problem of evaluating $E(M)$ using the log-exponential function [RW09, p. 27] with smoothing parameter ε. Similar to the basic k-means algorithm, *soft-k-means* clustering solves the stationarity conditions

$$\nabla_{m^j} E_\varepsilon(M) = 0, \quad j \in J \tag{3.6}$$

by fixed point iteration in terms of iteratively computing the **soft-assignments**

$$p^i_{\varepsilon,j}(M) = \frac{\exp\big(-D_{ij}(M)/\varepsilon\big)}{\sum_{k \in J} \exp\big(-D_{ik}(M)/\varepsilon\big)}, \quad q^j_{\varepsilon,i}(M) = \frac{p^i_{\varepsilon,j}(M)}{\sum_{k \in I} p^k_{\varepsilon,j}(M)}, \quad i \in I,\ j \in J \tag{3.7a}$$

with the so-called **mean shifts**

$$m^j = \sum_{i \in I} q^j_{\varepsilon,i}(M) x^i, \quad j \in J. \tag{3.7b}$$

The distributions

$$p^i_\varepsilon(M) \in \Delta_{|J|}, \quad i \in I \tag{3.8}$$

given by (3.7a) represent the soft-assignments $p^i_{\varepsilon,j}(M)$ of each data point x^i, $i \in I$ to each prototype m^j, $j \in J$, whereas the distributions

$$q^j_\varepsilon(M) \in \Delta_{|I|}, \quad j \in J \tag{3.9}$$

determine the convex combinations of data points that determine each prototype m^j by the mean shift (3.7b). Iterating the two steps (3.7) evolves the prototypes M until they reach a local minimum of the objective (3.5).

[1] The symbol 'k' is commonly used in the literature. We prefer in this paper however the more specific symbol J as index set for prototypes and use k (like i, j etc.) as free index.

3.2. **Divergence Functions and EM-Iteration.** An alternative and widely applied approach to clustering utilizes class-conditional distributions $p(x;\theta_j),\ j \in J$ and a corresponding mixture distribution

$$p(x;\Gamma) = \sum_{j\in J} \pi_j p(x;\theta_j) \tag{3.10}$$

as data model, with parameters

$$\Gamma = (\theta,\pi), \qquad \theta = (\theta_1,\dots,\theta_{|J|}), \qquad \pi = (\pi_1,\dots,\pi_{|J|})^\top \in \mathcal{S}, \tag{3.11}$$

with $\mathcal{S}$ defined by

$$\mathcal{S} = \{p \in \mathbb{R}^{|J|} \colon p_j > 0,\ j \in J,\ \langle \mathbb{1}, p\rangle = 1\}. \tag{3.12}$$

Clustering amounts to estimate the parameters Γ. Since the log-likelihood function corresponding to (3.10) is usually involved, maximizing a lower bound through the EM-iteration (EM: expectation-maximization) is the method of choice,

$$p(j|x^i;\Gamma^{(t)}) = \frac{\pi_j^{(t)} p(x^i;\theta_j^{(t)})}{\sum_{l\in J} \pi_l^{(t)} p(x^i;\theta_l^{(t)})}, \qquad j \in J \qquad \textbf{(E-step, soft-assignment)} \tag{3.13a}$$

$$\begin{aligned} \pi_j^{(t+1)} &= \frac{1}{|J|} \sum_{i\in I} p(j|x^i;\Gamma^{(t)}) \\ \theta_j^{(t+1)} &= \arg\max_{\theta_j} \sum_{i\in I} p(j|x^i;\Gamma^{(t)}) \log p(x^i;\theta_j) \end{aligned}, \qquad j \in J. \qquad \textbf{(M-step)} \tag{3.13b}$$

for some initialization $\Gamma^{(0)}$. We refer to [MP00] for background and further details.

Banerjee et al. [BMDG05] studied the case where the class-conditional distributions $p(x;\theta_j)$ of (3.10) belong to an exponential family of distributions [BN78] and, in particular, their representation in terms of a Bregman divergence function D_ϕ. Then the resulting data model (3.10) reads

$$p(x;\Gamma) = \sum_{j\in J} \pi_j \exp\big(-D_\phi(f(x),\eta_j)\big) b_\phi(x), \tag{3.14}$$

where f denotes a sufficient statistics regarded as feature vector, the factor b_ϕ accounts for normalization and $\eta_j = \nabla\psi(\theta_j)$ is determined by θ_j through conjugation of the convex log-partition function $\psi(\theta_j) = \log \int_{\mathcal{X}} p(x;\theta_j)\,\mathrm{d}x$. The corresponding EM-updates read

$$p(j|x^i;\Gamma^{(t)}) = \frac{\pi_j^{(t)} \exp\big(-D_\phi(f(x^i),\eta_j^{(t)})\big)}{\sum_{l\in J} \pi_l^{(t)} \exp\big(-D_\phi(f(x^i),\eta_l^{(t)})\big)}, \qquad j \in J \qquad \textbf{(E-step, soft-assignment)} \tag{3.15a}$$

$$\begin{aligned} \pi_j^{(t+1)} &= \frac{1}{|J|} \sum_{i\in I} p(j|x^i;\Gamma^{(t)}) \\ \eta_j^{(t+1)} &= \arg\min_{\eta_j} \sum_{i\in I} \nu_{j|i}(\Gamma^{(t)}) D_\phi\big(f(x^i),\eta_j^{(t)}\big) \end{aligned}, \qquad j \in J. \qquad \textbf{(M-step)} \tag{3.15b}$$

$$\nu_{j|i}(\Gamma^{(t)}) = \frac{p(j|x^i;\Gamma^{(t)})}{\sum_{k\in I} p(j|x^k;\Gamma^{(t)})}. \tag{3.15c}$$

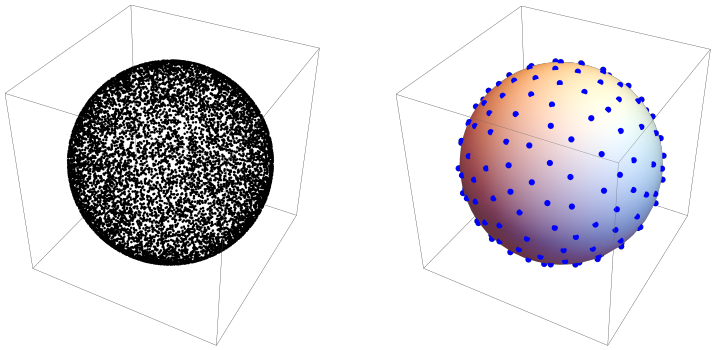

FIGURE 3.1. **Approximation of the metric clustering objective** (3.20)**.** LEFT: 10.000 points on the sphere regarded as manifold equipped with the cosine distance. RIGHT: 200 prototypes determined with linear runtime complexity by metric clustering are *almost uniformly located* in the data set, which qualifies them for unbiased initializations of computationally more involved nonlinear prototype evolutions. This works in any metric space, in particular on feature manifolds using the Riemannian distance or computationally less expensive divergence functions.

Moreover, since the Bregman divergence D_ϕ is induced by a convex function ϕ of Legendre type, the parameters η_j, $j \in j$ can be updated by the **mean-shifts**

$$\eta_j^{(t+1)} = \sum_{i \in I} \nu_{j|i}(\Gamma^{(t)}) f(x^i), \quad j \in J. \tag{3.16}$$

We exploit the above connection to divergence functions in Sections 4.1.2 and 4.3.2.

3.3. **Greedy-Based k-Center Clustering in Metric Spaces.** We adopt a simple algorithm from [HP11] as a *preprocessing step for data reduction*, due to the following properties. It works in any *metric space*

$$(X, d_X), \tag{3.17}$$

it has *linear* complexity $\mathcal{O}(|J||I|)$ with respect to the problem size $|I|$ which can be large, and it comes along with a *performance guarantee*: Given data points

$$X_I = \{x^1, \dots, x^{|I|}\} \subset X, \tag{3.18}$$

the objective is to determine a subset

$$M = \{m^1, \dots, m^{|J|}\} \subset X_I \tag{3.19}$$

that solves the combinatorially hard optimization problem

$$E_\infty^* = \min_{M \subset X_I, |M|=|J|} E_\infty(M), \qquad E_\infty(M) = \max_{x \in X_I} d_X(x, M), \tag{3.20}$$

where $d_X(x, M) = \min_{m \in M} d_X(x, m)$. Starting with a first initial point m^1, e.g. chosen randomly in X_I, selecting the remaining $|J| - 1$ points $m^2, \dots, m^{|J|}$ by greedy iteration yields a set M that is a 2-approximation $E_\infty(M) \leq 2E_\infty^*$ of the optimum (3.20) [HP11, Thm. 4.3]. As a consequence, the subset of $|J|$ points of M are almost uniformly distributed in X_I as measured by the metric d_X. Figure 3.1 provides an illustration.

We note that the greedy k-center clustering algorithm also works with a distance-like function, e. g. divergence function, instead of a metric d_X. We will later apply this algorithm in the context of clustering on a Riemannian manifold $\mathcal{M}$, where we have given a smooth (symmetric) divergence function D on $\mathcal{M}$. To

be more precise, we will use greedy k-center clustering as preprocessing in order to get an overcomplete set of labels as initial labels for the clustering approaches described in the next section.

4. Coupling Clustering on Manifolds and Spatially Regularized Assignment

We reformulate in Section 4.1 the iterative schemes of Sections 3.1 and 3.2 in order to cope with *manifold-valued data*. Both schemes will be coupled in Section 4.3 with the assignment flow that is presented in Section 4.2. This results in two novel schemes for spatially regularized label (prototype) learning from manifold-valued data. Finally, we define in Section 4.3.3 the *unsupervised assignment flow* as smooth interpolation of the flows corresponding to both schemes, depending on a single interpolation parameter.

4.1. Manifold-Valued Clustering.

We generalize the basic iterative clustering schemes of Sections 3.1 and 3.2 to manifold-valued data.

4.1.1. *Manifold-Valued Soft-k-Means Iteration.*

Let $(\mathcal{M}, g)$ be a smooth Riemannian manifold and let

$$\{z^1, \ldots, z^{|I|}\} \subset \mathcal{M} \tag{4.1}$$

be given data. We assume a smooth divergence function to be given (cf. Section 2.2)

$$D \colon \mathcal{M} \times \mathcal{M} \to \mathbb{R}, \qquad (x, y) \mapsto D(x, y) \tag{4.2}$$

that replaces the Riemannian distance d_g in order to compute Riemannian means more efficiently or even in closed form. We just use the symbol "D" and omit the subscript of (2.9), because what follows applies to a broad range of scenarios and to any corresponding concrete divergence function D. Examples are provided in Section 5.

We consider the task to determine a set of prototypes

$$M = \{m^1, \ldots, m^{|J|}\} \subset \mathcal{M} \tag{4.3}$$

by minimizing an objective function analogous to the soft-k-means objective (3.5),

$$E_\varepsilon(M) := E_\varepsilon(m^1, \ldots, m^{|J|}) = -\varepsilon \sum_{i \in I} \log \left(\sum_{j \in J} \exp\Big(-\frac{D(z^i, m^j)}{\varepsilon} \Big) \right), \quad \varepsilon > 0. \tag{4.4}$$

We next generalize the conditions (3.6). Let $d_j E_\varepsilon(M)$ denote the differential of the function $m^j \mapsto E_\varepsilon(m^1, \ldots, m^j, \ldots, m^{|J|})$. Then

$$d_j E_\varepsilon(M) = \sum_{i \in I} \underbrace{\frac{\exp\big(-\frac{D(z^i, m^j)}{\varepsilon} \big)}{\sum_{l \in J} \exp\big(-\frac{D(z^i, m^l)}{\varepsilon} \big)}}_{:= p^i_{\varepsilon, j}(M)} d_j D(z^i, m^j) = \sum_{i \in I} p^i_{\varepsilon, j}(M) d_j D(z^i, m^j), \quad j \in J, \tag{4.5}$$

where the **assignment probability vectors** $p^i_\varepsilon(M) \in \Delta_{|J|}$ play the same role as in eqns. (3.7a) and (3.8). They can be interpreted as *weight functions* depending on the prototypes M: setting temporarily $w_i = p^i_{\varepsilon, j}(M)$, $i \in I$, implies that eq. (4.5) has the same structure as the equation on the right of (2.6) after applying the differential on both sides, where we take into account that divergence functions $D(\cdot, \cdot)$ behave like squared distances (Section 2.2). Applying the mapping (2.3), we obtain the gradients and optimality conditions

$$(\operatorname{grad} E_\varepsilon)_j(M) = \widehat{g}^{-1}\big(d_j E_\varepsilon(M)\big) = \sum_{i \in I} p^i_{\varepsilon, j}(M) \widehat{g}^{-1}\big(d_j D(z^i, m^j)\big) = 0, \quad j \in J. \tag{4.6}$$

Comparing with (3.6) shows that, in the Euclidean case, the mean shift operation (3.7b) is defined by *normalized* weights $q^j_{\varepsilon, i}(M)$ due to (3.7a), conforming to the much more general situation (2.7). While normalization in (3.7a) is a *consequence* of the squared Euclidean distance of the objective (3.5), this may or

may not happen in (4.6), depending on the particular manifold $\mathcal{M}$, metric g and divergence function D at hand. Because subdiving each optimality condition (3.6) by the corresponding normalization factor in (3.7) does not change the condition, however, and because mean-shift on manifolds is performed with normalized weights, we define

$$p^i_{\varepsilon,j}(M) = \frac{\exp\big(-\frac{D(z^i,m^j)}{\varepsilon}\big)}{\sum_{l\in J}\exp\big(-\frac{D(z^i,m^l)}{\varepsilon}\big)}, \qquad q^j_{\varepsilon,i}(M) = \frac{p^i_{\varepsilon,j}(M)}{\sum_{k\in I} p^k_{\varepsilon,j}(M)}, \quad i\in I,\ j\in J \tag{4.7}$$

and in turn the **mean shift** (fixed point) **iteration**

$$(m^j)^{(t+1)} = \exp_{(m^j)^{(t)}}\Big(\sum_{i\in I} q^j_{\varepsilon,i}(M^{(t)})\widehat{g}^{-1}\big(d_j D((m^j)^{(t)}, z^i)\big)\Big), \quad j\in J \tag{4.8}$$

analogous to (2.8). Section 5 provides concrete examples for divergence functions on manifolds.

4.1.2. *Manifold-Valued EM-Iteration.* We consider again the situation (4.1)–(4.3) and adopt the clustering approach of Section 3.2. Iteration (3.15) generalizes to

$$p(j|z^i;M^{(t)}) = \frac{\pi_j^{(t)}\exp\big(-D(z^i,(m^j)^{(t)})\big)}{\sum_{l\in J}\pi_l^{(t)}\exp\big(-D(z^i,(m^l)^{(t)})\big)}, \qquad j\in J \qquad (\textbf{E-step, soft-assignment}) \tag{4.9a}$$

$$\begin{aligned}\pi_j^{(t+1)} &= \frac{1}{|J|}\sum_{i\in I} p(j|z^i;M^{(t)})\\ (m^j)^{(t+1)} &= \operatorname*{arg\,min}_{m^j}\sum_{i\in I}\nu_{j|i}(M^{(t)})D\big(z^i,(m^j)^{(t)}\big)\end{aligned}, \qquad j\in J. \qquad (\textbf{M-step}) \tag{4.9b}$$

$$\nu_{j|i}(M^{(t)}) = \frac{p(j|z^i;M^{(t)})}{\sum_{k\in I} p(j|z^k;M^{(t)})}. \tag{4.9c}$$

Note that we apparently ignore here the connection to class-conditional distributions $p(x;\theta_j)$ of the exponential family that formed the basis for the EM-iteration (3.15). This is not the case, however. Indeed, optimization problem (4.9b) which determines each prototype m^j by minimizing the *expected value* of a squared distance-like function, conforms to the updates (3.15b) and (3.16) of the *expectation parameter* $\eta_j = \nabla\psi(\theta_j) = \mathbb{E}_{\theta_j}[f_i]$, where the expectation is with respect to $p(x;\theta_j)$ and the sufficient statistics $f_i(x)$.

In order to solve problem (4.9b), we proceed analogously to (4.6). Examples with concrete choices of $D(\cdot,\cdot)$ are worked out in Section 5.

4.2. **Supervised Assignment Flow.** Let data be given by (4.1) together with *fixed* labels (prototypes) (4.3). The index set I corresponds to pixel locations $i\in I$ and extracted features z^i, $i\in I$, whereas the index set J enumerates the labels (class representatives, prototypes) m^j, $j\in J$. After fixing a suitable divergence function (4.2), the **distance vectors**

$$D_i(M) = \big(D(z^i,m^1),\ldots,D(z^i,m^{|J|})\big)\in\mathbb{R}^{|J|}, \quad i\in I \tag{4.10}$$

are defined. The approach [ÅPSS17] is based on the relatively open *probability simplex* of strictly positive vectors

$$\mathcal{S} = \{p\in\mathbb{R}^{|J|}\colon p_j>0,\ j\in J,\ \langle\mathbb{1},p\rangle = 1\} \tag{4.11}$$

with the uniform distribution as barycenter,

$$\mathbb{1}_{\mathcal{S}} := \frac{1}{|J|}\mathbb{1}_{|J|}, \qquad (\textbf{barycenter of } \mathcal{S}) \tag{4.12}$$

that becomes a Riemannian manifold when equipped with the *Fisher-Rao metric*

$$g_p(u,v) = \sum_{j\in J} \frac{u_j v_j}{p_j}, \quad u,v,\in T_0, \quad p \in \mathcal{S}, \tag{4.13}$$

where T_0 denotes the tangent space

$$T_0 := T_{\mathbb{1}_\mathcal{S}}\mathcal{S} = \{v \in \mathbb{R}^{|J|} \colon \langle \mathbb{1}, v\rangle = 0\}, \quad p \in \mathcal{S} \tag{4.14}$$

that we will work with throughout this paper, in lieu of the tangent spaces $T_p\mathcal{S} = \{u = \frac{v}{p} \colon v \in T_0,\ p \in \mathcal{S}\}$ that are equivalent up to the normalization. We denote by

$$R_p \colon \mathbb{R}^{|J|} \to T_0, \qquad d \mapsto R_p(d) = (\operatorname{Diag}(p) - pp^\top)d = p(d - \langle p, d\rangle \mathbb{1}), \quad p \in \mathcal{S} \tag{4.15}$$

a family of linear mappings onto T_0 parametrized by p.

Adopting the α-connection with $\alpha = 1$ from information geometry as introduced by Amari and Chentsov [AN00, Section 2.3], affine geodesics and a corresponding *exponential map* are given by

$$\operatorname{Exp} \colon \mathcal{S} \times T_0 \to \mathcal{S}, \qquad (p,v) \mapsto \operatorname{Exp}_p(v) := \frac{e^{\frac{v}{p}}}{\langle p, e^{\frac{v}{p}}\rangle}. \tag{4.16}$$

These geodesics are not length minimizing unlike the geodesics induced by the Riemannian (Levi-Civita) connection, but they closely approximate them [ÅPSS17, Prop. 3] and are computationally more convenient to work with. In particular, unlike exponential maps in general (cf. (2.5)), the map Exp_p is defined on the entire space T_0 and has the inverse [ÅPSS17, Appendix]

$$\operatorname{Exp}^{-1} \colon \mathcal{S} \times \mathcal{S} \to T_0, \qquad (p,q) \mapsto \operatorname{Exp}_p^{-1}(q) = R_p \log \frac{q}{p}, \tag{4.17}$$

with R_p given by (4.15). The composition of (4.16) and (4.15) defines the map[2]

$$\exp_p = \operatorname{Exp}_p \circ R_p \colon \mathbb{R}^{|J|} = T_0 \oplus \mathbb{R}\mathbb{1} \to \mathcal{S}, \qquad z \mapsto \frac{pe^z}{\langle p, e^z\rangle}, \qquad p \in \mathcal{S}. \tag{4.18}$$

These mappings apply to **assignment vectors**

$$W_i \in \mathcal{S}, \quad i \in I, \tag{4.19}$$

associated with each pixel $i \in I$ that represent the a posteriori probabilities

$$W_{ij} = \Pr(j|z^i), \quad i \in I,\ j \in J. \tag{4.20}$$

The assignment vectors form the row vectors of the **assignment matrix**

$$W = \begin{pmatrix} \vdots \\ W_i^\top \\ \vdots \end{pmatrix} = \begin{pmatrix} \cdots & W^j & \cdots \end{pmatrix} \in \mathcal{W} \subset \mathbb{R}_{++}^{|I|\times|J|}, \tag{4.21}$$

whose column vectors are denoted by W^j, $j \in J$. Due to (4.19), $W \in \mathcal{W}$ is regarded as point on the

$$\mathcal{W} = \mathcal{S} \times \cdots \times \mathcal{S} \qquad (|I| \text{ times}) \qquad (\textbf{assignment manifold}) \tag{4.22}$$

with tangent space

$$\mathcal{T}_0 = T_0 \times \cdots \times T_0 \qquad (|I| \text{ times}) \tag{4.23}$$

[2]With abuse of notation, this definition 'overloads' the symbol $\exp_p$ in connection with the manifold $\mathcal{S}$, in addition to the definition (4.18) for general manifolds $\mathcal{M}$. Due to the context, it will be always clear which definition applies.

and the corresponding mappings

$$\mathbb{1}_{\mathcal{W}} = (\mathbb{1}_{\mathcal{S}}, \ldots, \mathbb{1}_{\mathcal{S}}) \in \mathcal{W} \qquad \textbf{(barycenter)} \tag{4.24a}$$

$$R_W(Z) = \big(R_{W_1}(Z_1), \ldots, R_{W_{|I|}}(Z_{|I|})\big) \in \mathcal{T}_0, \qquad W \in \mathcal{W}, \quad Z \in \mathbb{R}^{|I|\times|J|} \tag{4.24b}$$

$$\operatorname{Exp}_W(V) = \big(\operatorname{Exp}_{W_1}(V_1), \ldots, \operatorname{Exp}_{W_{|I|}}(V_{|I|})\big) \in \mathcal{W}, \qquad W \in \mathcal{W}, \quad V \in \mathcal{T}_0 \tag{4.24c}$$

and $\exp_W, \operatorname{Exp}_W^{-1}$ similarly defined based on (4.18), (4.17).

We assume neighborhoods

$$\mathcal{N}_i = \{k \in I \colon ik \in \mathcal{E}\} \cup \{i\}, \quad i \in I \tag{4.25}$$

to be defined around each pixel $i \in I$, formally given by a graph $G = (I, \mathcal{E})$ with pixel indices I as vertex set and edges $\mathcal{E}$ defining (4.25). We associate with each neighborhood $\mathcal{N}_i$ weights $\{w_{ik} \colon k \in \mathcal{N}_i\}$ satisfying

$$w_{ik} > 0, \qquad \sum_{k\in\mathcal{N}_i} w_{ik} = 1, \qquad \forall i \in I. \tag{4.26}$$

These weights parametrize the regularization property of the assignment flow and are assumed to be given. We refer to [HSPS19] for an approach to learn these parameters from data. (4.25) and (4.26) define the *geometric mean* of assignment vectors [ÅPSS17, Lemma 5]

$$\mathcal{G}_i^w(W) = \operatorname{Exp}_{W_i}\Big(\sum_{k\in\mathcal{N}_i} w_{ik} \operatorname{Exp}_{W_i}^{-1}(W_k)\Big) = \exp_{W_i}\Big(\log \frac{\prod_{k\in\mathcal{N}_i} W_k^{w_{ik}}}{W_i}\Big), \qquad i \in I, \tag{4.27}$$

which defines – specifically for the assignment manifold (4.22) – a closed form solution to the general equation (2.8) that can be computed efficiently.

Using this setting, the assignment flow accomplishes image labeling as follows. Based on (4.10)

$$D = (D_1, \ldots, D_{|I|}) \in \mathbb{R}^{|I|\times|J|} \qquad \textbf{(distance vectors)} \tag{4.28}$$

are defined and mapped to

$$L(W) = \exp_W(D) \in \mathcal{W}, \qquad \textbf{(likelihood vectors)} \tag{4.29a}$$

$$L_i(W) := \frac{W_i e^{-\frac{1}{\rho}D_i}}{\langle W_i, e^{-\frac{1}{\rho}D_i}\rangle}, \quad \rho > 0, \qquad i \in I, \tag{4.29b}$$

where ρ is a user parameter to normalize the distances induced by the specific features f_i at hand. This representation of the data is regularized by geometric smoothing (4.27) to obtain the

$$S(W) \in \mathcal{W}, \qquad S_i(W) = \mathcal{G}_i^w\big(L(W)\big), \quad i \in I, \qquad \textbf{(similarity vectors)} \tag{4.30}$$

which in turn evolves the assignment vectors W_i, $i \in I$ through the

$$\dot{W} = R_W\big(S(W)\big), \qquad W(0) = \mathbb{1}_{\mathcal{W}}. \qquad \textbf{(assignment flow)} \tag{4.31}$$

We refer to [ÅPSS17] for further details and a discussion of the assignment flow (4.31): Each assignment vector $W_i(t) \in \mathcal{S}$ converges to an ε-neighborhood of some vertex (unit vector) $e_j \in \{0,1\}^{|J|}$, $j \in J$ of the closure $\overline{\mathcal{S}}$ and in this sense uniquely assigns a corresponding label $j \in J$ to each datum z^i, $i \in I$.

4.3. **Coupling the Assignment Flow and Label Evolution on Feature Manifolds.** We show in this section how combining the assignment flow (4.31) and the schemes of Section 4.1 results in *coupled flows* that *simultaneously* perform

- label evolution on a feature manifold, and
- spatially regularized label assignment to given data.

Coupling the assignment flow with the scheme of Section 4.1.1 defines the **coupled flow (CFa)** in Section 4.3.1, whereas coupling the assignment flow with the scheme of Section 4.1.2 defines the **coupled flow (CFb)** in Section 4.3.2. Comparing (CFa) and (CFb) in Section 4.3.3 shows that the latter flow subsumes the former one and hence defines the **unsupervised assignment flow**.

4.3.1. *Spatially Regularized Soft-k-Means on Feature Manifolds.* Minimizing the objective function (4.4) induces the assignment probilities

$$p^i_{\varepsilon,j}(M) = \frac{\exp\big(-\frac{1}{\varepsilon}D(z^i,m^j)\big)}{\sum_{l\in J}\exp\big(-\frac{1}{\varepsilon}D(z^i,m^l)\big)}, \quad i\in I,\ j\in J \tag{4.32}$$

due to (4.5). Regarding the assignment flow, the variables W_{ij} play the same role, see (4.20). The assignment flow (4.31) for W_{ij} reads

$$\dot W_{ij}(t) = W_{ij}(t)\Big(S_{ij}\big(W(t)\big) - \sum_{l\in J} W_{il}(t)S_{il}\big(W(t)\big)\Big), \quad i\in I,\ j\in J, \tag{4.33}$$

where the right-hand side comprises the similarity vectors $S_i(W)$, $i\in I$, whose j-th component due to (4.30), (4.27) and (4.29) is given by

$$S_{ij}(W) = \frac{\widetilde L_{\mathcal{N}_i,j}}{\langle \mathbb{1}, \widetilde L_{\mathcal{N}_i}\rangle}, \qquad \widetilde L_{\mathcal{N}_i,j} = \Big(\prod_{k\in\mathcal{N}_i} L_{kj}(W;M)\Big)^{w_{ik}} \tag{4.34a}$$

$$= \left(\prod_{k\in\mathcal{N}_i}\left(\frac{W_{kj}}{\big\langle W_k, e^{-\frac{1}{\rho}D_k(M)}\big\rangle}\right)^{w_{ik}}\right)\exp\Big(\sum_{k\in\mathcal{N}_i} w_{ik}\frac{D(z^k,m^j)}{\rho}\Big). \tag{4.34b}$$

This expression makes explicit how spatial regularization through averaging the given data (in terms of distance vectors) over local neighborhoods, is part of the vector field that drives the assignment flow. As a consequence, label assignments induced by $W(T)$, $T\gg 0$, are spatially more coherent.

Hence we propose to replace in (4.8) the normalized assignment variables $p^i_{\varepsilon,j}(M)$ given by (4.32), where *no* spatial regularization is involved, by the *normalized* assignment variables $q^j_{\varepsilon,i}(W)$ defined below by (4.35). The resulting **coupled flow (CFa)** that simultaneously performs label evolution and label assignment, reads

$$\textbf{(CFa)}\quad \begin{cases} \dot m^j(t) = -\alpha\displaystyle\sum_{i\in I} q^j_{\varepsilon,i}(W)\widehat{g}^{-1}\big(d_jD(z^i,m^j)\big), \quad m^j(0)=m^j_0, \quad \alpha>0, \quad j\in J, \\ \qquad\qquad q^j_\varepsilon(W) = \dfrac{W^j}{\langle\mathbb{1},W^j\rangle}, \quad j\in J \\ \dot W_i(t) = R_{W_i(t)}\big(S_i\big(W(t)\big)\big), \quad W_i(0)=\mathbb{1}_{\mathcal{S}}, \quad i\in I, \end{cases} \tag{4.35}$$

with W^j due to (4.21) and user parameter α that enables the adjust the time scale of the label flow induced by $\dot m^j(t)$, $j\in J$ relative to the assignment flow induced by $\dot W_i(t)$, $i\in I$.

4.3.2. *Spatially Regularized EM-Iteration on Feature Manifolds.* The scheme of Section 4.1.2 and the update formulas (4.9) suggest an alternative coupling of label evolution and the assignment flow. Equation (4.29b) reads

$$L_{ij}(W;M) = \frac{W_{ij} e^{-\frac{1}{\rho} D(z^i, m^j)}}{\sum_{l \in J} W_{il} e^{-\frac{1}{\rho} D(z^i, m^l)}}, \tag{4.36}$$

which agrees with the right-hand side of (4.9a), except for the scaling parameter ρ and the assignment variables W_{ij} in place of the mixture coefficients π_j. Indeed, since there is *no* interaction between different spatial locations $i \in I$ on the right-hand side of (4.36), $L_{ij}(W_i; M)$ can be interpreted as *local* posterior probability of label j given the observation z^i, in agreement with the left-hand side of (4.9a). Likewise, applying the first update equation of (4.9b) to (4.36) yields

$$W_{ij}^{(t+1)} = \frac{1}{|I|} \sum_{i \in I} L_{ij}(W^{(t)}; M^{(t)}), \quad j \in J \tag{4.37}$$

which does *not* depend on $i \in I$. We therefore take into account spatial regularization by replacing the mixture coefficients π_j, $j \in J$ by the variables W_{ij}, $i \in I, j \in J$, that are governed by the assignment flow and hence *do* spatially interact. The resulting **coupled flow (CFb)** reads

$$\textbf{(CFb)} \quad \begin{cases} \dot m^j(t) = -\alpha \sum_{i \in I} \nu_{j|i}\big(W(t), M(t)\big) \widehat g^{-1}\big(d_j D(z^i, m^j(t))\big), \quad m^j(0) = m_0^j,\ \alpha > 0,\ j \in J, \\ \qquad\qquad \nu_{j|i}(W, M) = \dfrac{L_{ij}(W;M)}{\sum_{k \in I} L_{kj}(W;M)}, \quad L_{ij}(W;M) = \dfrac{W_{ij} e^{-\frac{1}{\rho} D(z^i, m^j)}}{\sum_{l \in J} W_{il} e^{-\frac{1}{\rho} D(z^i, m^l)}}, \\ \dot W_i(t) = R_{W_i(t)}\big(S_i\big(W(t)\big)\big), \quad W_i(0) = \mathbb{1}_{\mathcal{S}}, \quad i \in I, \end{cases} \tag{4.38}$$

where $W(t)$ depends on M through (4.34).

4.3.3. *Unsupervised Assignment Flow.* We examine the relation between the coupled flows (CFa) (4.35) and (CFb) (4.38). Comparing $L_{ij}(W;M)$ given by (4.38) with $q_{\varepsilon,i}^j(W)$ given by (4.35) shows due to $\sum_{l \in J} W_{il} = 1,\ i \in I$ and

$$W_{ij} = \lim_{\rho \to \infty} L_{ij}(W;M) \tag{4.39a}$$

that

$$q_{\varepsilon,i}^j(W) = \lim_{\rho \to \infty} \nu_{j|i}(W, M). \tag{4.39b}$$

We conclude that **(CFa) is a special case of (CFb)**. Since the scaling parameter ρ plays a unique role in (4.29), however, we propose to parametrize $L_{ij}(W;M)$ of (4.38) in the *same* way, but with another *independent* parameter $\sigma > 0$ replacing ρ, in order to 'interpolate' smoothly between the coupled flows (CFa) and (CFb) in the sense of (4.39).

As a result, the final form of our approach, called **unsupervised assignment flow (UAF)**, reads

$$\textbf{(UAF)} \quad \begin{cases} \dot m^j(t) = -\alpha \sum_{i \in I} \nu_{j|i}\big(W(t), M(t)\big) \widehat g^{-1}\big(d_j D(z^i, m^j(t))\big), \quad m^j(0) = m_0^j, \quad \alpha > 0,\ j \in J, \\ \qquad\qquad \nu_{j|i}(W, M) = \dfrac{L_{ij}^\sigma(W;M)}{\sum_{k \in I} L_{kj}^\sigma(W;M)}, \quad L_{ij}^\sigma(W;M) = \dfrac{W_{ij} e^{-\frac{1}{\sigma} D(z^i, m^j)}}{\sum_{l \in J} W_{il} e^{-\frac{1}{\sigma} D(z^i, m^l)}}, \quad \sigma > 0, \\ \dot W_i(t) = R_{W_i(t)}\big(S_i\big(W(t)\big)\big), \quad W_i(0) = \mathbb{1}_{\mathcal{S}}, \quad i \in I, \end{cases} \tag{4.40}$$

with user parameters $\alpha > 0$ controlling the relative speed of label vs. assignment evolution, and parameter $\sigma > 0$ as just discussed.

As already mentioned in Section 3.3, the greedy k-center clustering provides an overcomplete set of labels in a preprocessing step which is used as initial condition $\{m_0^j\}_{j\in J}$ for the prototype component of the unsupervised assignment flow (4.40).

4.4. **Geometric Numerical Integration.** In this subsection, we detail the iterative scheme that was used in Section 5 for numerically integrating the coupled unsupervised assignment flow (4.40). We rewrite these equations more compactly in the form

$$\dot{W}(t) = R_{W(t)} F_{\mathcal{W}}\big(W(t), M(t)\big), \quad W(0) = \mathbb{1}_{\mathcal{W}}, \tag{4.41a}$$

$$\dot{M}(t) = F_{\mathcal{M}^{|J|}}\big(W(t), M(t)\big), \quad M(0) = M_0, \tag{4.41b}$$

where the dependency of $F_{\mathcal{W}}(W, M) = S(W)$ on M is implicitly given via the distance vectors (4.10) and the dependency of the similarity vectors on these distance vectors – cf. (4.29) and (4.30).

In order to uniformly evaluate our approach for various feature manifolds $\mathcal{M}$, we simply use the Riemannian *explicit* Euler scheme for integrating the prototype evolution flow (4.41b), i. e.,

$$M^{(t+1)} = \exp_{M^{(t)}}\Big(h F_{\mathcal{M}^{|J|}}\big(W^{(t)}, M^{(t)}\big)\Big), \tag{4.42}$$

with step size $h > 0$, and $\exp_{M^{(t)}}$ is defined by (2.5) for the Riemannian manifold $\mathcal{M}^{|J|} = \mathcal{M} \times \cdots \times \mathcal{M}$. In order to numerically integrate the assignment flow (4.41a), we adapt the geometric *implicit* Euler scheme from [ZSPS18]. It amounts to solving the fixed point equation

$$V^{(t+1)} = h \Pi_{\mathcal{T}_0} F_{\mathcal{W}}\big(\exp_{W^{(t)}}\big(V^{(t+1)}\big), M^{(t+1)}\big)\big), \tag{4.43}$$

by an iterative inner loop, where $\Pi_{\mathcal{T}_0}$ denotes the orthogonal projection onto the tangent space (4.23), followed by updating

$$W^{(t+1)} = \exp_{W^{(t)}}\big(V^{(t+1)}\big). \tag{4.44}$$

Here, $\exp_W$ denotes the map given by (4.18) and (4.24).

5. Case Studies: Label Learning on Feature Manifolds

In the preceding section, we derived the unsupervised assignment flow (4.40) for a general feature manifold $\mathcal{M}$ together with a geometric numerical integration scheme. In the following three subsections, we illustrate the approach by working out details of three concrete feature manifolds. These scenarios will be evaluated numerically in the experiments section 6.

5.1. $\mathrm{SO}(3)$**-Valued Image Data: Orthogonal Frames in** $\mathbb{R}^3$**.** In this subsection, we study clustering on the Lie group $\mathrm{SO}(n)$ of $n \times n$ rotation matrices. This is a smooth Riemannian manifold whose tangent space at $R \in \mathrm{SO}(n)$ is given by

$$T_R\mathrm{SO}(n) = \{R\Omega \colon \Omega \in \mathfrak{so}(n)\}, \tag{5.1}$$

where $\mathfrak{so}(n) = \{\Omega \in \mathbb{R}^{n\times n} \colon \Omega^\top = -\Omega\}$ denotes the Lie algebra of $\mathrm{SO}(n)$, and with the Riemannian metric given by the Frobenius inner product $g_R(A_1, A_2) = \mathrm{tr}(A_1^\top A_2)$. Based on the matrix exponential expm and logarithm logm [Hig08], the corresponding exponential and logarithmic maps read

$$\exp_R(R\Omega) = R\,\mathrm{expm}(\Omega), \quad \log_{R_1}(R_2) = R_1\,\mathrm{logm}(R_1^\top R_2), \tag{5.2}$$

and the Riemannian distance is given by

$$d_{\mathrm{SO}(n)}(R_1, R_2) = \|\,\mathrm{logm}(R_1^\top R_2)\|_F. \tag{5.3}$$

In the specific case $n = 3$, well known formulas in closed form are available [Hig08]. By Rodrigues' formula, the matrix exponential of $A \in \mathfrak{so}(3)$ is given by

$$\operatorname{expm}(A) = I + \operatorname{sinc}(a)A + \tfrac{1}{2}\operatorname{sinc}^2(\tfrac{a}{2})A^2, \qquad a = \sqrt{\tfrac{1}{2}\operatorname{tr}(A^\top A)}, \tag{5.4}$$

with the sinc-function

$$\operatorname{sinc}(x) = \begin{cases} \frac{\sin(x)}{x}, & x \neq 0 \\ 1, & x = 0. \end{cases} \tag{5.5}$$

The matrix logarithm of $R \in \mathrm{SO}(3)$ with $\operatorname{tr}(R) = 1 + 2\cos(\theta)$, $|\theta| < \pi$ is given by

$$\operatorname{logm}(R) = \frac{1}{2\operatorname{sinc}(\theta)}(R - R^\top). \tag{5.6}$$

Moreover, the Riemannian distance can be evaluated without computing the matrix logarithm or an eigenvalue decomposition as

$$d_{\mathrm{SO}(3)}(R_1, R_2) = \sqrt{2}\arccos\left(\frac{\operatorname{tr}(R_1^\top R_2) - 1}{2}\right), \quad R_1, R_2 \in \mathrm{SO}(3). \tag{5.7}$$

Regarding the clustering of data $\{R_i\}_{i\in I} \subset \mathrm{SO}(n)$, we use the canonical divergence function $D(R_1, R_2) = \frac{1}{2}d_{\mathrm{SO}(n)}(R_1, R_2)^2$. As a result, the flow of the **(UAF)** (4.40) for the prototypes $S_j \in \mathrm{SO}(n)$ takes the form

$$\dot{S}_j(t) = \alpha \sum_{i\in I} \nu_{j|i}\big(W(t), S(t)\big) \operatorname{Log}_{S_j(t)}(R_i) = \alpha \sum_{i\in I} \nu_{j|i}\big(W(t), S(t)\big) S_j(t) \operatorname{logm}(S_j(t)^\top R_i), \quad j \in J. \tag{5.8}$$

Discretizing this flow due to (4.42) yields the multiplicative update scheme

$$S_j^{(t+1)} = S_j^{(t)} \operatorname{expm}\left(\alpha h \sum_{i\in I} \nu_{j|i}^{(t)} \operatorname{logm}\left(\big(S_j^{(t)}\big)^\top R_i\right)\right), \quad j \in J. \tag{5.9}$$

5.2. **Orientation Vector Fields.** We consider the task to cluster two-dimensional orientation vector fields that we regard as maps from the image domain into the Grassmann manifold $\mathcal{M} = \mathrm{Grass}(1, 2)$. We identify $\mathrm{Grass}(1, 2) \cong \mathbb{R}/\pi\mathbb{Z}$, i. e., we identify the subspace $\{\lambda(\cos\theta, \sin\theta)^\top : \lambda \in \mathbb{R}\} \subset \mathbb{R}^2$ and the angle $\theta \in [0, \pi)$. Let $q\colon \mathbb{R} \to \mathbb{R}/\pi\mathbb{Z}$ be the quotient map $\theta \mapsto \theta \mod \pi$. Rather than operating directly on the quotient manifold $\mathbb{R}/\pi\mathbb{Z}$, we work with representatives of its elements in $\mathbb{R}$. In particular, a flow on the quotient manifold will be given by $q(\vartheta(t))$, where $\vartheta(t)$ is a flow in $\mathbb{R}$. For any two representatives $\theta_1, \theta_2 \in \mathbb{R}$, the induced distance is given by

$$d(\theta_1, \theta_2) = d_{\mathcal{M}}(q(\theta_1), q(\theta_2)) = \min_{\varphi\in\pi\mathbb{Z}} |\theta_1 - \theta_2 + \varphi| \in [0, \tfrac{\pi}{2}], \tag{5.10}$$

and we have $d(\theta_1, \theta_2) = 0$ if and only if $q(\theta_1) = q(\theta_2)$. For the unsupervised assignment flow (4.40), we choose the canonical divergence function $D(x, y) = \frac{1}{2}\big(d_{\mathcal{M}}(x, y)\big)^2$ on $\mathcal{M}$. By (5.10), this corresponds to the dissimilarity function $D(\theta_1, \theta_2) = \frac{1}{2}\big(d(\theta_1, \theta_2)\big)^2$ for representatives $\theta_1, \theta_2 \in \mathbb{R}$. This dissimilarity function is differentiable if $q(\theta_1) \neq q(\theta_2 + \frac{\pi}{2})$, i. e., if the minimizer $\varphi^* \in \arg\min_{\varphi\in\pi\mathbb{Z}} |\theta_1 - \theta_2 + \varphi|$ is unique. In this case, we have $\frac{\partial}{\partial\theta_2} D(\theta_1, \theta_2) = \theta_2 - \theta_1 - \varphi^*$. Now, denoting $\{\theta_i\}_{i\in I} \subset \mathbb{R}$ the representatives of given orientations at pixels $i \in I$ and denoting by $\{\vartheta_j\}_{j\in J} \subset \mathbb{R}$ the representatives of the prototype orientations (labels), the label evolution of (4.40) takes the form

$$\dot{\vartheta}_j(t) = \alpha \cdot \left(\sum_{i\in I} \nu_{j|i}\big(W(t), \vartheta(t)\big)\big(\theta_i - \varphi_{ij}^*(t)\big) - \vartheta_j(t)\right) \quad \text{with} \quad \varphi_{ij}^*(t) \in \operatorname*{arg\,min}_{\varphi\in\pi\mathbb{Z}} \big|\theta_i - \vartheta_j(t) + \varphi\big|. \tag{5.11}$$

Since this flow evolves in $\mathbb{R}^{|J|}$, it can be numerically integrated using classical integration schemes. As mentioned above, the corresponding prototype flow in $\mathcal{M} \cong \mathbb{R}/\pi\mathbb{Z}$ then is given by $q(\vartheta_j(t))$, $j \in J$.

5.3. **Feature Covariance Descriptors Fields.** We consider data given as covariance region descriptors, as introduced in [TPM06]. Details of the corresponding unsupervised assignment flow are worked out in Section 5.3.1. In Section 5.3.2, we generalize the representation to obtain descriptors that are invariant with respect to rotations of the image domain.

5.3.1. *Basic Approach.* We consider feature maps $f\colon I \to \mathbb{R}^s$ extracted from a given 2D image $u\colon I \to \mathbb{R}^c$ with c channels by taking partial derivatives channel-wise, e. g. $u_x = \frac{\partial u}{\partial x}$ and $u_{xy} = \frac{\partial^2 u}{\partial x \partial y}$. A typical example used in our experiments is

$$i \mapsto f^i = \Big(u, u_x, u_y, u_{xx}, \sqrt{2}u_{xy}, u_{yy}\Big)^\top \in \mathbb{R}^{6c} \tag{5.12}$$

where $(x, y)^\top$ denote the image coordinates at pixel $i \in I$. The corresponding covariance descriptor C_i with respect to a pixel neighborhood $\mathcal{N}(i) \subset I$ is given by

$$C_i = \sum_{j \in \mathcal{N}(i)} \omega_{ij}(f^j - \bar{f}^i)(f^j - \bar{f}^i)^\top + \varepsilon \mathrm{Id} \quad \text{with} \quad \bar{f}^i = \sum_{j \in \mathcal{N}(i)} \omega_{ij} f^j, \quad 0 < \varepsilon \ll 1, \tag{5.13}$$

where $\omega_i = (\omega_{ij})_{j \in \mathcal{N}_i} \in \Delta_{|\mathcal{N}(i)|}$ are weights. We add the identity matrix with a very small ε to ensure that all descriptors are positive definite, which otherwise may not hold in particular cases like homogeneous "flat" image regions.

Now we consider the task of clustering given covariance descriptors as points on the Riemannian manifold of symmetric positive definite matrices [Bha06]

$$\mathcal{P}_s = \big\{X \in \mathbb{R}^{s \times s}\colon X = X^\top,\ X \text{ is positive definite}\big\} \tag{5.14}$$

endowed with the Riemannian metric $g_X(U, V) = \operatorname{tr}(X^{-1}UX^{-1}V)$ on each tangent space $T_X\mathcal{P}_s = \big\{U \in \mathbb{R}^{s \times s}\colon U^\top = U\big\}$. The Riemannian gradient of a function $F\colon \mathcal{P}_s \to \mathbb{R}$ is given by $\operatorname{grad} F(X) = X\partial F(X)X \in T_X\mathcal{P}_s$, where ∂F denotes the Euclidean gradient of F. Denoting the prototypes (labels) by $\{\Lambda_j\}_{j \in J} \subset \mathcal{P}_s$, the label flow of (4.40) reads

$$\dot{\Lambda}_j(t) = -\alpha \sum_{i \in I} \nu_{j|i}\big(W(t), \Lambda(t)\big)\Lambda_j(t)\partial_2 D\big(C_i, \Lambda_j(t)\big)\Lambda_j(t), \quad j \in J, \tag{5.15}$$

where $D(X, Y)$ is a proper divergence on $\mathcal{P}_s$ as discussed below, and $\partial_2 D(X, Y)$ denotes its Euclidean gradient with respect to Y. In the following, we discuss possible choices of D. An obvious choice is the canonical divergence induced by the Riemannian distance

$$D_{\mathrm{R}}(X, Y) = \frac{1}{2} d_{\mathcal{P}_s}(X, Y)^2 = \frac{1}{2} \sum_{k \in [s]} \big(\log \lambda_k(X, Y)\big)^2, \tag{5.16}$$

which involves all generalized eigenvalues $\lambda_k(X, Y)$ of the matrix pencil (X, Y). Considering that $D(C_i, \Lambda_j)$ has to be computed for each pair of datum C_i and prototype Λ_j at *each point of time* when integrating the flow, the computation of the generalized eigenvalues would be very expensive computationally. As a more efficient alternative to D_{R}, we consider the Stein divergence [Sra13]

$$D_{\mathrm{S}}(X, Y) = \log\det\left(\tfrac{X+Y}{2}\right) - \tfrac{1}{2}\log\det(XY), \tag{5.17a}$$

$$\partial_2 D_{\mathrm{S}}(X, Y) = \frac{1}{2}\left(\left(\tfrac{X+Y}{2}\right)^{-1} - Y^{-1}\right). \tag{5.17b}$$

It involves the determinant and the inverse of a positive definite matrix, which both can be efficiently computed using the Cholesky decomposition. Based on the choice $D = D_S$, equation (5.15) takes the form

$$\dot\Lambda_j(t) = \frac{\alpha}{2}\Big(\Lambda_j(t) - \Lambda_j(t)Q_j(t)\Lambda_j(t)\Big) \quad \text{with} \quad Q_j(t) = \sum_{i\in I} \nu_{j|i}\big(W(t),\Lambda(t)\big)\left(\frac{C_i+\Lambda_j(t)}{2}\right)^{-1}. \tag{5.18}$$

Taking the exponential map $\exp_X(U) = X^{\frac12}\operatorname{expm}\big(X^{-\frac12}UX^{-\frac12}\big)X^{\frac12}$ into account, with expm denoting the matrix exponential, and discretizing the flow with the Riemannian explicit Euler scheme (4.42), gives the prototype update for the Stein divergence

$$\Lambda_j^{(t+1)} = \tilde\Lambda_j \operatorname{expm}\left(\frac{\alpha h}{2}\Big(I - \tilde\Lambda_j Q_j^{(t)}\tilde\Lambda_j\Big)\right)\tilde\Lambda_j \quad \text{with} \quad \tilde\Lambda_j = \Big(\Lambda_j^{(t)}\Big)^{\frac12}. \tag{5.19}$$

5.3.2. *Rotational Invariance.* We additionally constructed a dissimilarity function on $\mathcal{P}_s$ that is invariant under rotations of the image domain. In contrast to the Stein divergence, this dissimilarity function takes the special structure of covariance descriptors into account and hence depends on the underlying feature map. We consider the feature map in (5.12) as example.

Let $u, \tilde u\colon \mathbb{R}^2 \to \mathbb{R}$ denote two grayvalue images that are related by an Euclidean transformation

$$\begin{pmatrix}\tilde x\\ \tilde y\end{pmatrix} = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}\begin{pmatrix}x\\ y\end{pmatrix} + \begin{pmatrix}x_0\\ y_0\end{pmatrix}, \quad \theta\in[0,2\pi) \tag{5.20}$$

of the image domain, i. e. $\tilde u(\tilde x,\tilde y) = u(x,y)$. Their derivatives transform as

$$\begin{pmatrix}\tilde u_x\\ \tilde u_y\end{pmatrix} = R_1(\theta)\begin{pmatrix}u_x\\ u_y\end{pmatrix}, \qquad \begin{pmatrix}\tilde u_{xx}\\ \sqrt{2}\tilde u_{xy}\\ \tilde u_{yy}\end{pmatrix} = R_2(\theta)\begin{pmatrix}u_{xx}\\ \sqrt{2}u_{xy}\\ u_{yy}\end{pmatrix}, \tag{5.21}$$

with rotation matrices $R_1(\theta)\in \mathrm{SO}(2)$ and $R_2(\theta)\in\mathrm{SO}(3)$ given by

$$R_1(\theta) = \begin{pmatrix}\cos\theta & -\sin\theta\\ \sin\theta & \cos\theta\end{pmatrix}, \qquad R_2(\theta) = \begin{pmatrix}\cos^2\theta & -\sqrt{2}\cos\theta\sin\theta & \sin^2\theta\\ \sqrt{2}\cos\theta\sin\theta & \cos^2\theta-\sin^2\theta & -\sqrt{2}\cos\theta\sin\theta\\ \sin^2\theta & \sqrt{2}\cos\theta\sin\theta & \cos^2\theta\end{pmatrix}. \tag{5.22}$$

It follows that covariance descriptors of $u\colon I\to\mathbb{R}^c$ with the feature map (5.12) transform as $\tilde C = R(\theta)CR(\theta)^\top$, with a rotation matrix $R(\theta)\in\mathrm{SO}(s)$. Setting

$$\mathcal{R} \coloneqq \big\{R(\theta)\colon \theta\in[0,2\pi)\big\}, \tag{5.23}$$

it turns out that $\mathcal{R}$ is a one-dimensional subgroup of $SO(s)$, i. e. $R(\theta_1+\theta_2) = R(\theta_1)R(\theta_2)$. Eventually, we construct the rotation-invariant dissimilarity function by minimizing over the Lie group action of $\mathcal{R}$, i. e.

$$D_{\mathrm{S},\mathcal{R}}(X,Y) \coloneqq \min_{R\in\mathcal{R}} D_{\mathrm{S}}\big(X, RYR^\top\big) = \min_{R\in\mathcal{R}} D_{\mathrm{S}}\big(R^\top XR, Y\big). \tag{5.24}$$

If $\partial_2 D_{\mathrm{S}}(R^{*\top}XR^*,Y) = (R^{*\top}XR^*+Y)^{-1} - \frac12 Y^{-1}$ is the same for all $R^*\in\arg\min_{R\in\mathcal{R}} D_{\mathrm{S}}\big(R^\top XR,Y\big)$, then $D_{\mathrm{S},\mathcal{R}}(X,Y)$ is differentiable in Y and the derivative is given by $\partial_2 D_{\mathrm{S},\mathcal{R}}(X,Y) = \partial_2 D_{\mathrm{S}}(R^{*\top}XR^*,Y)$ [BS13, Theorem 4.13+Remark 4.14]. This holds in particular if R^* is unique for a given pair (X,Y).
Using the divergence $D_{\mathrm{S},\mathcal{R}}$, equation (5.15) yields the same prototype update formulas (5.18) and (5.19) as for the Stein divergence, except for the modification

$$Q_j(t) = \sum_{i\in I}\nu_{j|i}\big(W(t),\Lambda(t)\big)\left(\frac{R_{ij}(t)^\top C_i R_{ij}(t) + \Lambda_j(t)}{2}\right)^{-1} \tag{5.25}$$

with $R_{ij}(t)\in\arg\min_{R\in\mathcal{R}} D_{\mathrm{S}}\big(R^\top C_i R, \Lambda_j(t)\big)$.

Remark. We conclude this section with further comments on the invariant dissimilarity function (5.24).

(1) We point out again that $\mathcal{R}$ due to (5.23) (and its existence) depends on the feature map f. For the specific case (5.12) considered above, a transformation of the form $\tilde{C} = R(\theta)CR(\theta)^\top$ exists since *all* derivatives up to a given order are involved. Furthermore, $\mathcal{R}$ is a subgroup of $\mathrm{SO}(s)$ due to the proper normalization of the mixed derivatives (note the factor $\sqrt{2}$).
(2) Evaluating (5.24) amounts to solve a one-dimensional smooth but non-convex problem. We omit the details.
(3) The dissimilarity function $D_{\mathrm{S},\mathcal{R}}$ is not a divergence function as introduced in Section 2.2, since $D_{\mathrm{S},\mathcal{R}}(X,Y) = 0$ does not imply $X = Y$, but only $[X]_\mathcal{R} = [Y]_\mathcal{R}$ with $[X]_\mathcal{R} = \{RXR^\top : R \in \mathcal{R}\}$. Unfortunately, this cannot be fixed by considering the quotient $\mathcal{P}_s/_\sim$ with $X \sim Y$ if and only if $X \in [Y]_\mathcal{R}$, since $\mathcal{P}_s/_\sim$ does not have a manifold structure (e.g. the equivalence class of the identity matrix is a singleton). Nevertheless, we can plug-in $D_{\mathrm{S},\mathcal{R}}$ into our approach that can be used with any differentiable dissimilarity function. The resulting prototypes are then representatives $\{\Lambda_j\}_{j\in J} \subset \mathcal{P}_s$ of classes $[\Lambda_j]_\mathcal{R}$.
(4) The set of pairs $(X,Y) \in \mathcal{P}_s \times \mathcal{P}_s$, for which $D_{\mathrm{S},\mathcal{R}}(X,Y)$ is not differentiable, is negligible [RW09, Theorem 10.31]. But even for such pairs one can choose some optimal $R_{ij}(t)$ in (5.25), such that the prototype flow remains well-defined.

6. Numerical Examples

In this section, we demonstrate and compare the proposed **unsupervised assignment flow (UAF)** using several synthetic and real-world images and different feature manifolds, as detailled in Section 5. As described in Section 4.4, the geometric numerical integration of the **(UAF)** was carried out using the geometric implicit Euler scheme for the assignment component of the flow and a Riemannian explicit Euler scheme for the prototype component of the flow. For both schemes, we used the fixed step-size $h = 0.1$ in all experiments. Additionally we adopted in our implementation the renormalization step from [ÅPSS17] with $\varepsilon = 10^{-10}$ for the assignment component, to avoid numerical issues for assignments very close to the boundary of simplex $\Delta_{|J|} = \overline{\mathcal{S}}$. Uniform weights (w_{ik}) were used for regularizing the assignments through geometric averaging (4.27). The integration process terminated when the average entropy of the assignment component dropped below 10^{-3} which indicates almost unique assignments (probability vectors are close to unit-vectors) and in turn that the weights $\nu_{j|i}(W,M)$ for the prototype evolution become stationary as well. We initialized the assignment component of the unsupervised assignment flow with the uninformative barycenter (all labels are equiprobable). The initial prototypes were determined by greedy k-center metric clustering as discussed in Section 3.3, in order to obtain an almost uniformly sampled dictionary from the input data. The number of labels $|J|$ was chosen large enough to start with an overcomplete dictionary.

6.1. Parameter Influence.

This experiment discusses the influence of the two model parameters σ and α of the **(UAF)** as defined by (4.40). Parameter σ determines the trade-off between the influence of the assignments (spatial regularization) and the influence of the distances in the feature space on the weights $\nu_{j|i}(W,M)$ which govern the label evolution. $\sigma = \infty$ results in the coupled flow **(CFa)** where the weights $\nu_{j|i}(W,M)$ solely depend on the assignments, whereas $\sigma = \rho$ gives coupled flow **(CFb)** which incorporates both, the spatially regularized assignment and the distances in the feature space, into the dictionary update step. In general, the impact of spatial regularization on the evolution of labels evolution decreases with decreasing values of σ, and the influence of the distances in feature space on the evolution of labels is even stronger for $\sigma < \rho$.

Parameter α controls the relative speed of the evolution of labels vs. assignments. If α is set too small, i. e., if the evolution of labels is too slow, then hardly any label evolution occurs at all during the period the assignment evolution so that the resulting assignment is effectively comparable to the *supervised* assignment

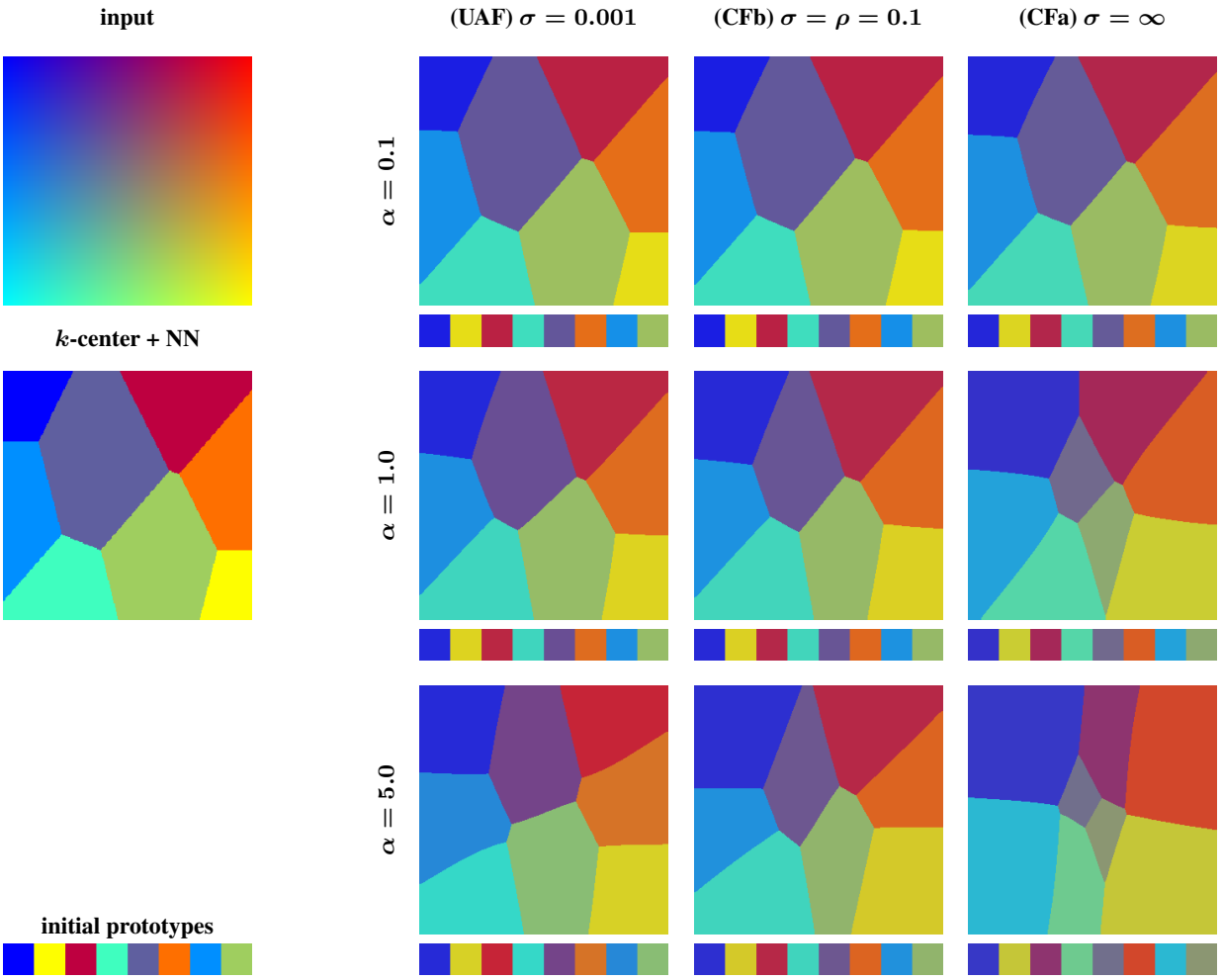

FIGURE 6.1. **Influence of the parameters σ and α.** The figure illustrates the influence of the parameters σ and α on the unsupervised assignment flow **(UAF)** in terms of the resulting labelings. From the smooth input image (left panel, top), the initial prototypes ($|J| = 8$) are extracted by greedy k-center clustering and assigned by the nearest neighbor (NN) rule. The right panel shows the labelings returned by the **(UAF)**, for different values of σ and α, after termination of the coupled evolution of labels and assignments. We observe for increasing values σ and α that regions are "attracted" towards the center of the image domain, since label colors are increasingly averaged through the spatially regularized assignments.

flow [ÅPSS17] based on the initial set of labels. On the contrary, if α is set too large, labels adapt too fast to the current assignment, which may be undesirable especially if the assignment is still too close to the uninformative barycenter in the initial phase of its evolution.

In order to visualize clearly the role of σ and α, we consider in this section the RGB color space as feature space. The demonstrated effects carry over to the other non-trivial feature manifolds, of course. We used a $|\mathcal{N}| = 3 \times 3$ neighborhood size for geometric spatial regularization and fixed the number of labels to $|J| = 8$.

Figure 6.1 illustrates the above discussion for an academical computer-generated color image with a smooth strong gradient, which was generated such that from left-to-right the red channel is increasing and

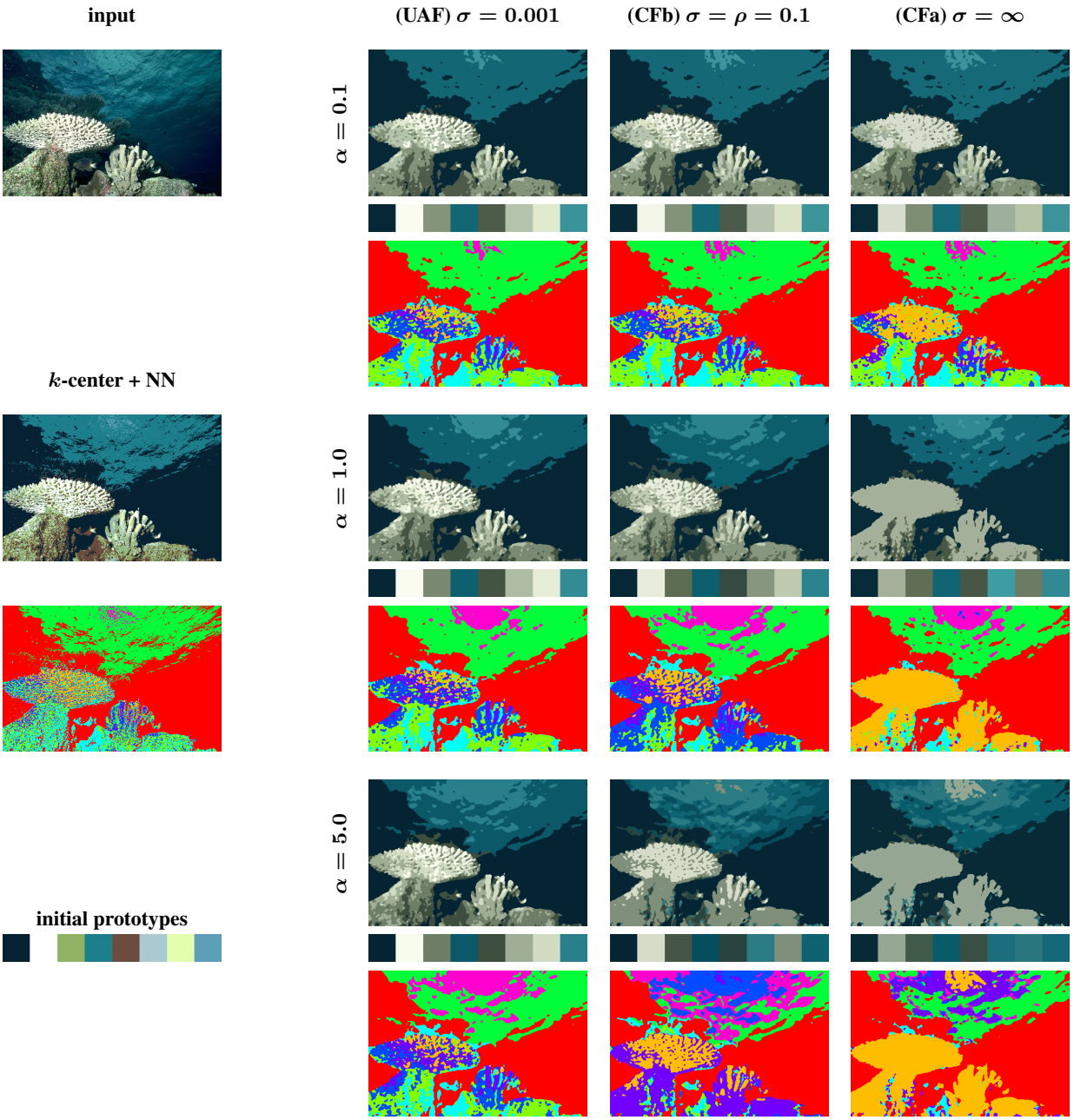

FIGURE 6.2. **Influence of the parameters σ and α.** Results of the **(UAF)** are shown that reproduce for a real image the effects illustrated by Figure 6.1. Each labeling is additionally shown using false colors to ease the perception of differences. We observe for increasing σ an increasing impact of spatial regularization, whereas for increasing α labels adapt faster along with the size of the spatially regularized regions.

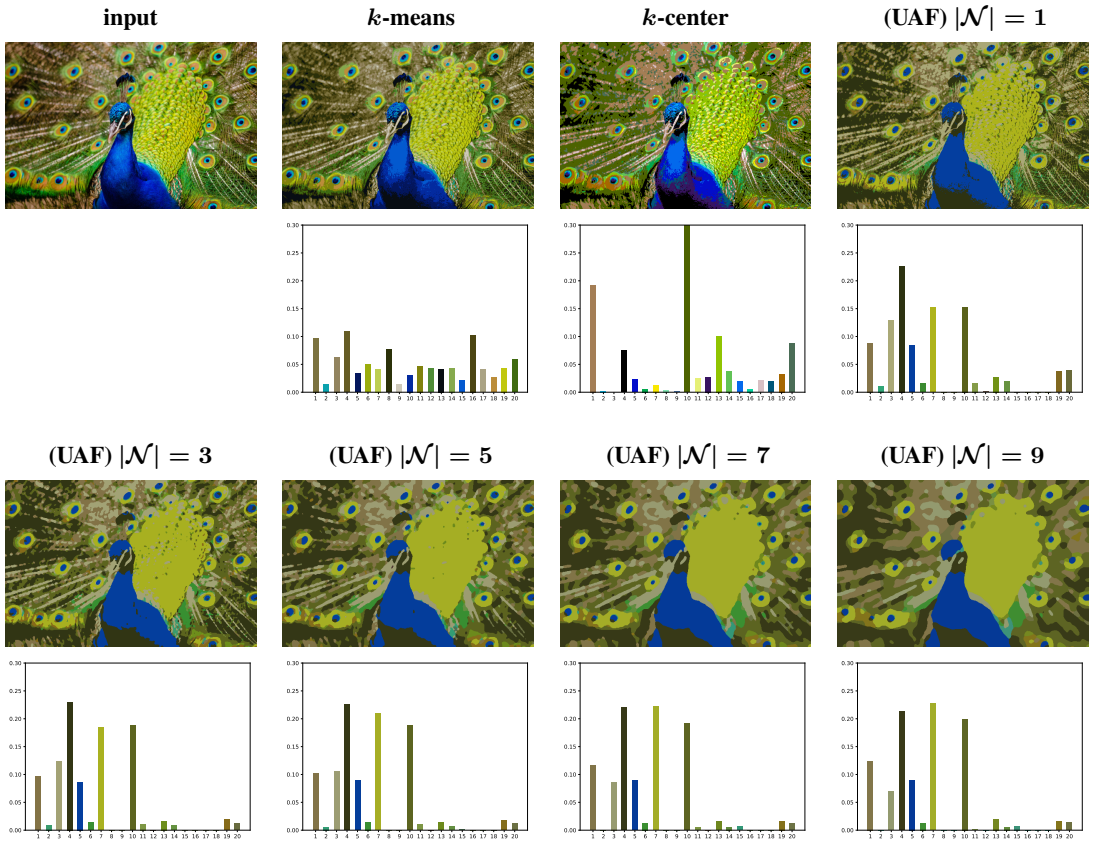

FIGURE 6.3. **Effect of spatial regularization.** We compare the proposed **(UAF)** to k-means clustering and k-center clustering, respectively, and demonstrate the effect spatial regularization, parametrized by increasing neighborhood sizes used for geometric averaging, on the resulting label statistics and label assignments. The histogram bars are colored by the corresponding labels, and their heights indicate the relative amount of assigned pixels. We observe that as the scale (neighborhood size) of spatial regularization increases, the label set quickly becomes more sparse.

the blue channel is decreasing, whereas the green channel is increasing from top to bottom. The increasing ability of labels to adapt (increasing α) and the increasing impact of spatial regularization, is illustrated by the cell sizes of the final Voronoi diagram relative to the initial configuration.

Figure 6.2 demonstrates the same effects for a real image. The partitions corresponding to the unsupervised image labelings are additionally displayed using false colors in order to highlight the differences. The interpretation of the results for different values of σ and α is analogous to the effects shown by Figure 6.1.

Specifically, we observe that for a small value $\sigma = 0.001$ (column **(UAF)**), which increases the influence of the distances in the feature space, the resulting labeling preserves fine scales (e.g. see left coral in Figure 6.2) in comparison to the other extreme choice $\sigma = \infty$ (column **(CFa)**), where the influence of the spatial regularization through the assignments in the image domain is maximal and hence fine scales are removed from the resulting labeling. The intermediate parameter choice $\sigma = \rho = 0.1$ (column **(CFb)**) shows a good compromise between the effects caused by the two extreme values of σ.

The influence of parameter α controlling the relative speed of label and assignment evolution can be seen row-wise. For small $\alpha = 0.1$, the adaption of the prototypes is quite limited. For the choice $\alpha = 1.0$, we observe a good compromise between label evolution and spatial regularization through the assignment flow. Finally, a very large value $\alpha = 5.0$ results in strong spatial regularization, since the labels are adapting relatively fast to the current assignments and consequently the regions assigned to labels grow faster.

6.2. **Effect of Spatial Regularization.** Figure 6.3 illustrates the effect of spatial regularization performed by the **(UAF)** on the evolution of both labels and label assignments, by comparing to basic k-means clustering and to k-center clustering (Section 3.3), respectively, where no spatial regularization is involved at all. The parameter values $\alpha = 1.0$ and $\sigma = \infty$ were used.

Comparing k-means with k-center clustering shows that k-means clustering selects a more uniform quantization for the feature data, whereas the greedy k-center clustering rather picks more extremal points in

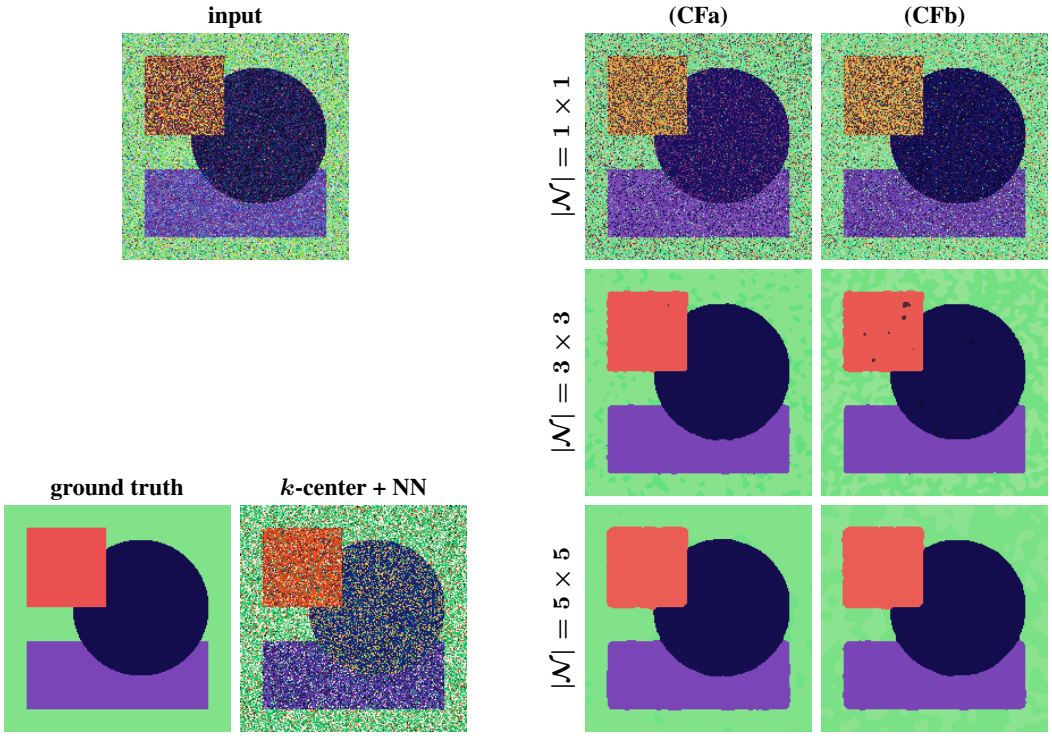

FIGURE 6.4. **Unsupervised label learning for $\mathbf{SO(3)}$-valued image data.** Rotation matrices are color coded by the scheme adopted from [KMBB15]. Each label (orthogonal frame, rotation matrix) is also depicted as trihedron by Figure 6.5 using as background the false color used here. The input data were generated from ground truth as described in the text. The panel 'k-center + NN' depicts the nearest neighbor assignments of 8 initial labels selected from the input data by metric clustering (Section 3.3). Panels on the right depict both the labels and the assignment of these labels by the two versions **(CFa)** and **(CFb)** of the unsupervised assignment flow **(UAF)**. Spurious labels "die out" and, for a reasonably large neighborhood size used for spatial regularization, high-quality labelings are determined simultaneously. The resulting labels are visualized by Figure 6.5.

the feature space which subsequently serve as initial prototypes for **(UAF)**. The remaining panels demonstrate that spatial regularization quickly sparsifies the label set as the scale (neighborhood size) of spatial regularization increases.

6.3. **Case Studies: Label Learning on Feature Manifolds.** In this section, we demonstrate the "plug in and play" property of the unsupervised assignment flow **(UAF)** by applying it to the scenarios worked out in Section 5. In principle, *any* Riemannian feature manifold can be used provided a corresponding divergence function $D(\cdot,\cdot)$ and the exponential map admit a computational feasible evaluation of the **(UAF)** through the numerical scheme (4.42).

We next consider the scenarios of Section 5 in turn.

6.3.1. $\mathrm{SO}(3)$*-Valued Image Data: Orthogonal Frames in* $\mathbb{R}^3$. Figure 6.4 depicts ground truth data in terms of orthogonal frames assigned to each pixel $i \in I$ and visualized with false colors. Each ground-truth label is also shown as trihedron by Figure 6.5.

The input data (Fig. 6.4) were generated by independently sampling for each pixel $i \in I$ a vector $n_i \sim \mathcal{N}(0, \sqrt{0.5} I_3)$, determining a corresponding random skew-symmetric matrix $\Omega(n_i) \in \mathfrak{so}(3)$, and by replacing the ground-truth value R_i by $R_i \exp(\Omega(n_i))$.

Next we determined by metric clustering (Section 3.3) an overcomplete set of $|J| = 8$ initial prototypes as shown by Figure 6.5. The corresponding nearest neighbor (NN) assignments are shown by Figure 6.4. They clearly illustrate the need for spatially regularized assignments, *not only* for determining a reasonably coherent partition of the image domain *but also* for affecting label evolution, in order to determine proper labels enabling to find such a partition by assignment.

The labelings generated by unsupervised assignment flow **(UAF)** are shown by Figure 6.4, for the parameters $\sigma = \rho = 1.0$ and $\sigma = \infty$ corresponding to the specific versions **(CFa)** and **(CFb)** of the **(UAF)**, and using different neighborhood sizes $|\mathcal{N}| \in \{1 \times 1, 3 \times 3, 5 \times 5\}$ for spatial regularization. The relative speed parameter α for the prototype evolution flow was set to the natural value $\alpha = 1$ (cf. Section 6.1). The results show that, for both flows **(CFa)** and **(CFb)**, spurious labels "die out" whereas the remaining labels converge

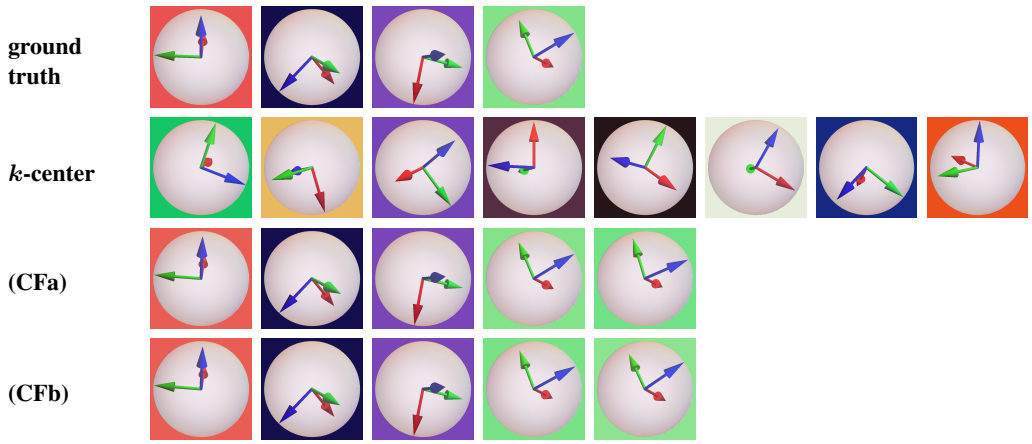

FIGURE 6.5. **Label visualization for $\mathrm{SO}(3)$ data.** Each label corresponding to the results depicted by Figure 6.4 is shown here as trihedron, using the false color in Fig. 6.4 as background color here. Three labels of the initial label set (second row) "died out" during the unsupervised assignment flow evolution, whereas the remaining ones converged to values quite close to ground truth.

to values quite close to ground truth (Fig. 6.5). Specifically, for the large green background region, two labels close to the ground truth label are recovered due to the initial fluctuations within a large spatial region.

We point out that the only essential parameter value required for a reasonable result is the scale (neighborhood size) of spatial regularization.

6.3.2. *Orientation Vector Fields.* Given a gray-scale image (Figure 6.6) we estimated orientations of local image structure from local gradient scatter matrices. Orientations are encoded at each pixel by the angle between the horizontal axis and the smallest eigenvector. The resulting data take values in $\mathbb{R}/_{\pi\mathbb{Z}} \cong S^1$ after identifying antipodal points. Figure 6.6 shows the nearest neighbor assignments of the initial $|J| = 8$ prototypes determined by greedy k-center clustering from the noisy input data, together with labels and label assignments of the versions **(CFa)** and **(CFb)** of the unsupervised assignment flow **(UAF)** corresponding to the parameter choices $\sigma = \rho = 0.1$ and $\sigma = \infty$. The relative speed parameter α for the prototype evolution was set to $\alpha = 0.5$ and $|\mathcal{N}| = 5 \times 5$ neighborhoods were used for spatial averaging.

Both flows managed to position a label correctly in the neighborhood of $0 \cong \pi$ (visualized in red) and only required seven labels to properly encode the data by labeling.

6.3.3. *Feature Covariance Descriptor Fields.* We demonstrate the application of the unsupervised assignment flow to the manifold of positive definite matrices. For a given input image, we extracted the covariance descriptor using the feature map (5.12) and $|\mathcal{N}| = 5 \times 5$ in (5.13). We applied version **(CFa)** of the unsupervised assignment flow to a synthetic and a real world image, i. e. setting $\sigma = \infty$ ensuring a strong effect of spatial regularization on label evolution. Initial sets of $|J| = 10$ labels were determined by metric clustering, to ensure interpretation of the results visualized by false colors. Due to the higher dimension of the feature space of this scenario, a larger value $\alpha = 10$ of the relative speed parameter controlling the prototype evolution turned out to be useful for both test instances.

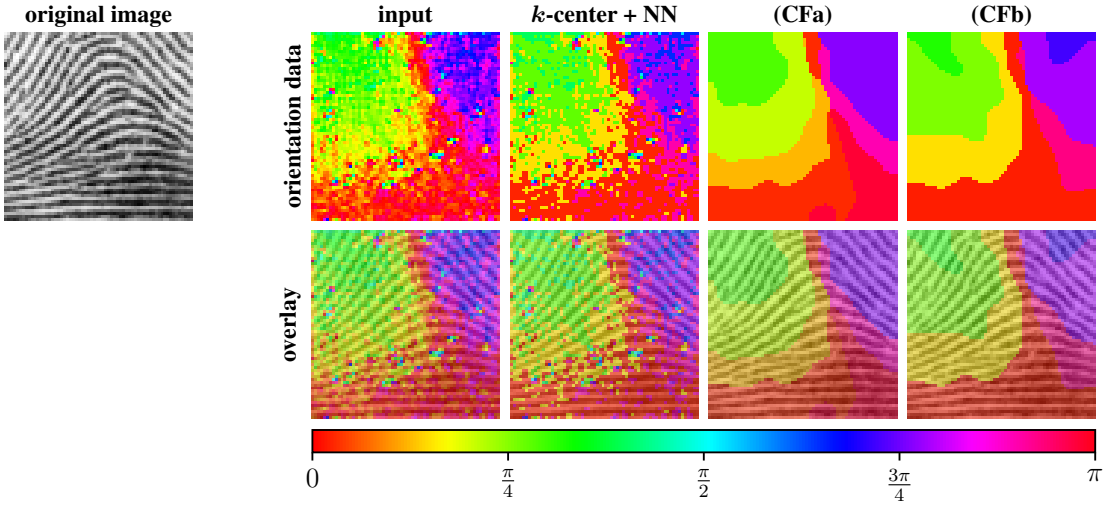

FIGURE 6.6. **Unsupervised label learning from orientation vector fields.** Orientations are extracted from the gray-scale image, using the spectral decomposition of local scatter matrices of the image gradient, and represented as elements of $\mathbb{R}/_{\pi\mathbb{Z}} \cong S^1$ as described in the text. Using a correspoding distance function, the unsupervised assignment flow learns both proper labels, including their number, and label assignments for encoding the noisy input data.

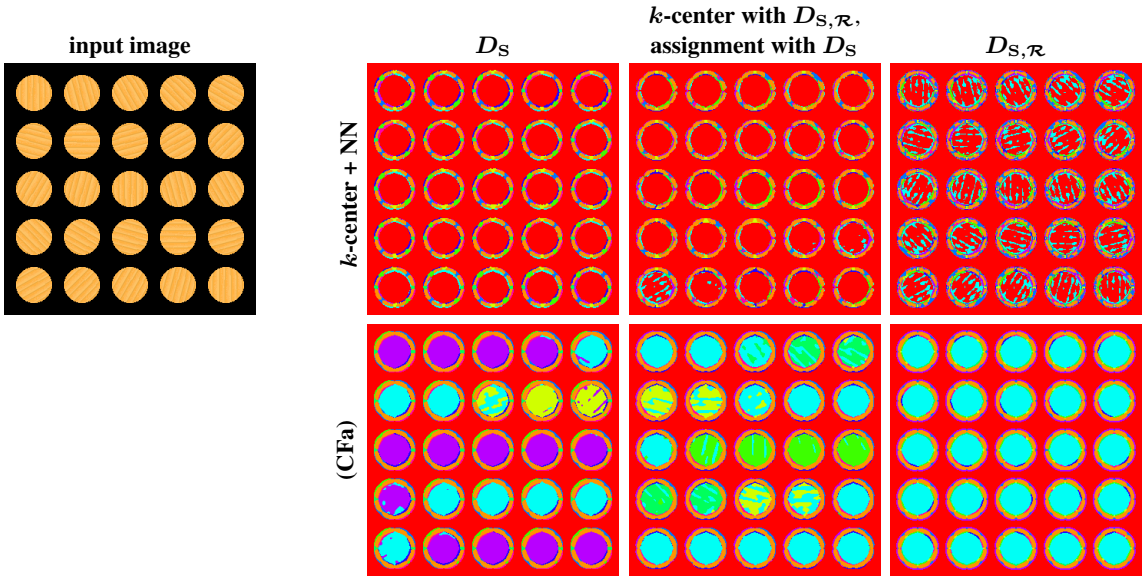

FIGURE 6.7. **Unsupervised learning of rotationally invariant labels from covariance descriptors.** The input data are covariance descriptors (5.13) extracted from the input image which comprises a texture rotated in steps of 15 degrees. Both labels and label assignments are pixelwise visualized using false colors in the panels on the right (only color differences matter, rather than the colors themselves). The unsupervised assignment flow **(CFa)** together with the rotationally invariant Stein divergence $D_{\mathrm{S},\mathcal{R}}$ returns a small set of labels that encodes local image structure irrespective of its orientiation. By contrast, using the Stein divergence D_{S} is less effective.

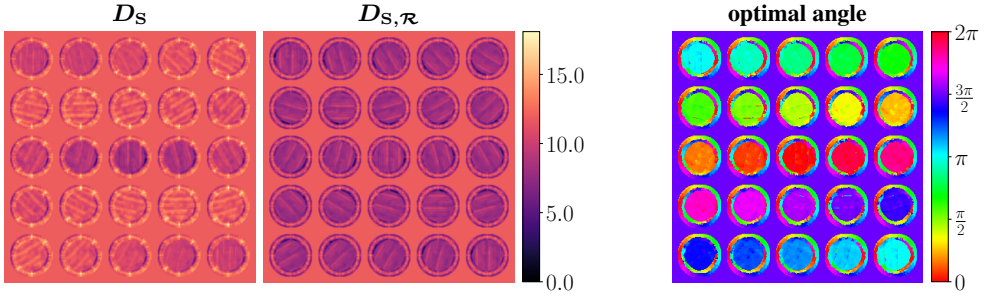

FIGURE 6.8. **Comparing the Stein divergence D_{S} with its rotationally invariant variant $D_{\mathrm{S},\mathcal{R}}$.** Using the covariance descriptors illustrated by Figure 6.7, the panels on the left show pixelwise the distances to some fixed (arbitrary) label. Contrary to the uniform distances $D_{\mathrm{S},\mathcal{R}}$, the distance D_{S} strongly depends on the orientation of the texture. On the right-hand side, the optimal rotation angles are shown corresponding to the evaluation of $D_{\mathrm{S},\mathcal{R}}$. These angles accurately recover rotations of the texture.

Figure 6.7 depicts a synthetic image with a texture rotated in steps of 15 degrees. $|\mathcal{N}| = 3 \times 3$ neighborhoods were used for spatial averaging and the constant of (5.13) was set to $\varepsilon = 10^{-5}$ to ensure strict positive definiteness even in completely homogeneous regions of this computer-generated image. Initial prototypes

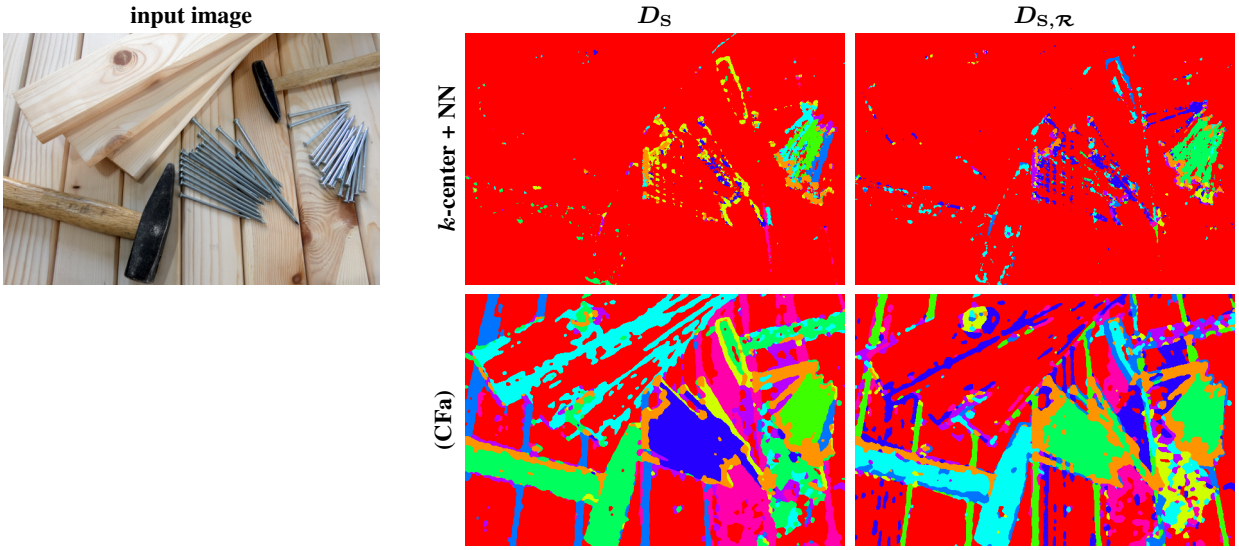

FIGURE 6.9. **Unsupervised learning of covariance descriptor labels through label assignment.** The depicted results were obtained for the real input image on the left and are analogous to the results of the synthetic scenario depicted by Figure 6.7. The local assignments of initial labels (top row on the right) highlight that metric clustering completely ignores the spatial structure of the input data. The results returned by the unsupervised assignment flow **(CFa)**, therefore, are impressive (bottom row): labels and label assignments jointly evolve so as to capture the spatial image structure. While the distance D_{S} is sensitive to orientations of texture, the distance $D_{\mathrm{S},\mathcal{R}}$ is not: the final labels and label assignments (bottom right) basically partition the image into wooden texture independent of the orientation of the wooden boards (encoded with red), nails and similar line structures in the background (encoded with green), the hammers (light-blue) and oriented wooden texture (blue), independent of the local orientation of these textures.

were extracted from the input data using the greedy metric k-center clustering using the Stein divergence D_{S} and its rotation-invariant version $D_{\mathrm{S},\mathcal{R}}$, respectively. The experiments below should not only demonstrate another feature manifold that can be flexibly handled using the proposed unsupervised assignment flow, but they should also assess if numerical results display the rotational invariance of $D_{\mathrm{S},\mathcal{R}}$ that holds by construction mathematically (Section 5.3.2).

The six panels on the right of Figure 6.7 show columnwise the results of local label assignments (k-center + NN) and the assignments after label evolution performed by **(CFa)**, respectively, using either distance D_{S} or $D_{\mathrm{S},\mathcal{R}}$. Regarding the results depicted by the center column, greedy k-center clustering was performed using $D_{\mathrm{S},\mathcal{R}}$, while the nearest neighbor (NN) assignment and **(CFa)** were performed using D_{S}, in order to highlight the difference between D_{S} and $D_{\mathrm{S},\mathcal{R}}$ based on the same initial prototypes.

The result show using that $D_{\mathrm{S},\mathcal{R}}$ leads to an unsupervised labeling of all textures with a single label only. Thus, depending on the application, using $D_{\mathrm{S},\mathcal{R}}$ instead of the basic Stein divergence D_{S} can lead to more compact label dictionaries determined by the proposed unsupervised assignment flow. Figure 6.8 underlines this finding from a different angle. The two panels on the left display *pixelwise* the distances D_{S} and $D_{\mathrm{S},\mathcal{R}}$ from some fixed (arbitrary) reference descriptor. The two images show quantitatively that D_{S} is highly non-uniform, unlike $D_{\mathrm{S},\mathcal{R}}$. The panel on the right of Figure 6.8 visualizes for each pixel the optimal angle minimizing (5.24) over (5.23), that has to be determined for the evaluation of $D_{\mathrm{S},\mathcal{R}}$. One can clearly see

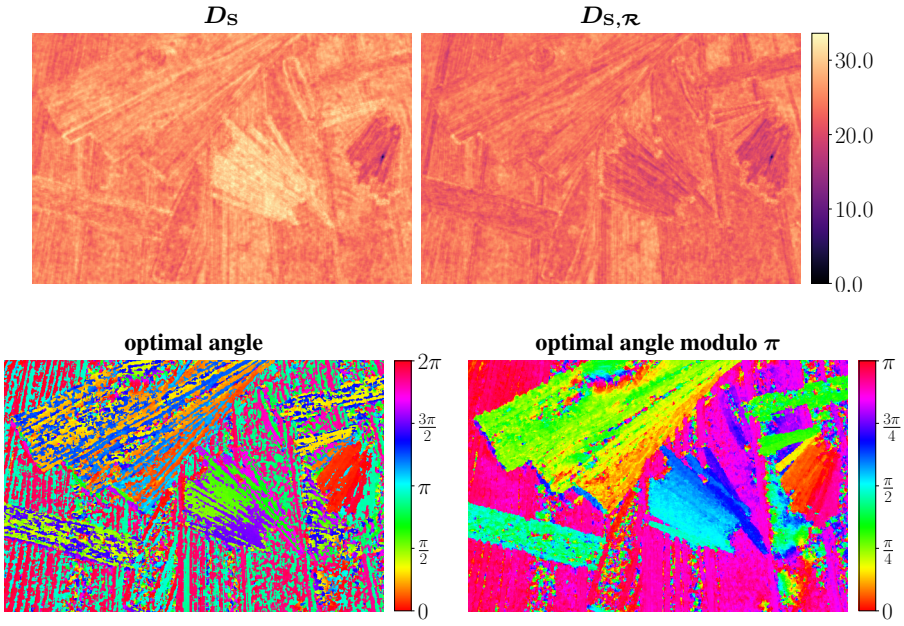

FIGURE 6.10. **Comparing the Stein divergence $D_{\mathbf{S}}$ with its rotationally invariant variant $D_{\mathbf{S},\boldsymbol{\mathcal{R}}}$.** The depicted results correspond to the scenario of Figure 6.9 and are analogous to the results shown by Figure 6.8 for the synthetic scenario illustrated by Figure 6.7. The top row shows the pixelwise distances between each covariance descriptors extracted from the image of Figure 6.9 and a fixed prototype located at the pile of nails on the right. While the distance D_{S} considerably differs between two piles of nails due to the different orientations, the rotationally-invariant distance $D_{\mathrm{S},\mathcal{R}}$ is more uniform. The bottom row displays pixelwise the optimal rotation angle that determines $D_{\mathrm{S},\mathcal{R}}$. Up to unavoidable local errors of these locally computed estimates, the distance $D_{\mathrm{S},\mathcal{R}}$ recovers the local orientation of the real texture in the input image (bottom right).

how the rotations of the textures of the input image of Figure 6.7 are recovered. This may be useful for some applications as well.

Figure 6.9 depicts a real world image. We used $|\mathcal{N}| = 5 \times 5$ neighborhoods for spatial averaging and $\varepsilon = 10^{-7}$ for the constant of (5.13) to ensure strict positive definiteness of the covariance descriptors. Analogous to Figure 6.7, we compared the nearest neighbor (NN) assignment and the result returned by **(CFa)** with respect to the Stein divergence D_{S} and its rotationally invariant version $D_{\mathrm{S},\mathcal{R}}$, respectively.

We observe that the rotationally invariant feature representation together with the unsupervised assignment flow ($D_{\mathrm{S},\mathcal{R}}$ / **(CFa)**; panel bottom-right) leads to an unsupervised label representation of the input data that basically partitions the image into wooden texture independent of the orientation of the wooden boards (encoded with red), nails and similar line structures in the background (encoded with green), the hammers (light-blue) and oriented wooden texture (blue).

Analogous to Figure 6.8, Figure 6.10 (first row) shows the pixel-wise distances to a fixed label (located at the right pile of nails) for the distances D_{S} and $D_{\mathrm{S},\mathcal{R}}$, respectively. Comparing the distances to the two piles of nails illustrates once again and quantitatively the rotational invariance of $D_{\mathrm{S},\mathcal{R}}$. The bottom row of panels shows the corresponding optimal rotation angles corresponding to the evaluation of $D_{\mathrm{S},\mathcal{R}}$, as defined by (5.24). These angles recover the relative orientation of the textures which may be useful for some applications.

7. Conclusion

We proposed the unsupervised assignment flow for performing jointly label evolution on feature manifolds and spatially regularized label assignment to given feature input data. The approach alleviates the requirement for supervised image labeling to have proper labels at hand, because an initial set of labels can evolve and adapt to better values while being assigned to given data.

The derivation of our approach highlights that it encompasses related state-of-the-art approaches to unsupervised learning: soft k-means clustering and EM-based estimation of mixture distributions with distributions of the exponential family as mixture components (class-conditional feature distributions). We generalized these approaches to manifold-valued data and defined the unsupervised assignment flow by coupling label evolution with the assignment flow adopted from [ÅPSS17]. We suggested greedy k-center clustering for determining an initial label set that works with linear complexity in any metric space and with fixed approximation error bounded from above, for every application.

The separation between feature evolution and spatial regularization through assignments enables the flexible application of our approach to various scenarios, provided some key operations (divergence function evaluation, exponential map) are computational feasible for the particular feature manifold at hand. We demonstrated this property for three different scenarios and showed that coupling the evolution of labels and assignments has beneficial effects in either direction. The approach involved two parameters whose role is well understood. As a consequence, the only essential parameter is the neighborhood size used for spatial regularization.

Our unsupervised learning approach is consistent in that the very same approach that is used for supervised labeling is used for label learning, without need to resort to approximate inference due to the complexity of learning, as is the case e.g. for learning with graphical models.

A key property of our approach is the sparsifying effect of spatial assignment regularization on unsupervised label learning. Our future work will study this property in connection with label learning from the assignment flow itself, in terms of patches of assignments at coarser spatial scales. Furthermore, all experiments in this paper were conducted using uniform weights $(w_{ik})_{k\in\mathcal{N}_i}$ for the spatial regularization of assignments (cf. Eq. (4.27)). Learning these weights from data in order to represent the spatial context of typical feature occurrences as prior knowledge has been studied recently [HSPS19]. Working out a mathematically consistent way to extend this approach to unsupervised scenarios, as studied in the present paper, defines an exciting modeling problem.

References

[AC10] S.-I. Amari and A. Cichocki, *Information geometry of divergence functions*, Bull. Pol. Acad. Sci.: Tech. **58** (2010), no. 1, 183–195.

[AJLS17] N. Ay, J. Jost, H. V. Lê, and L. Schwachhöfer, *Information Geometry*, Springer, 2017.

[AN00] S.-I. Amari and H. Nagaoka, *Methods of Information Geometry*, Amer. Math. Soc. and Oxford Univ. Press, 2000.

[ÅPSS17] F. Åström, S. Petra, B. Schmitzer, and C. Schnörr, *Image Labeling by Assignment*, J. Math. Imaging Vision **58** (2017), no. 2, 211–238.

[Bas13] M. Basseville, *Divergence measures for statistical data processing – An annotated bibliography*, Signal Proc. **93** (2013), no. 4, 621–633.

[BB97] H. H. Bauschke and J. M. Borwein, *Legendre Functions and the Method of Random Bregman Projections*, J. Convex Analysis **4** (1997), no. 1, 27–67.

[Bha06] R. Bhatia, *Positive Definite Matrices*, Princeton Univ. Press, 2006.

[BMDG05] A. Banerjee, S. Merugu, I. S. Dhillon, and J. Ghosh, *Clustering with Bregman Divergences*, J. Mach. Learn. Res. **6** (2005), 1705–1749.

[BN78] O. E. Barndorff-Nielsen, *Information and Exponential Families in Statistical Theory*, Wiley, Chichester, 1978.

[BS13] J Frédéric Bonnans and Alexander Shapiro, *Perturbation analysis of optimization problems*, Springer Science & Business Media, 2013.

[CM02] D. Comaniciu and P. Meer, *Mean Shift: a Robust Approach Toward Feature Space Analysis*, IEEE Trans. Patt. Anal. Mach. Intell. **24** (2002), no. 5, 603–619.

[CS16] A. Cherian and S. Sra, *Positive Definite Matrices: Data Representation and Applications to Computer Vision*, Algorithmic Advances in Riemannian Geometry and Applications (H. Minh and V. Murino, eds.), Springer, 2016, pp. 93–114.

[CSBP13] A. Cherian, S. Sra, A. Banerjee, and N. Papanikolopoulos, *Jensen-Bregman LogDet Divergence with Application to Efficient Similarity Search for Covariance Matrices*, IEEE PAMI **35** (2013), no. 9, 2161–2174.

[CZ97] Y. A. Censor and S. A. Zenios, *Parallel Optimization: Theory, Algorithms, and Applications*, Oxford Univ. Press, New York, 1997.

[FH75] K. Fukunaga and L. Hostetler, *The Estimation of the Gradient of a Density Function, with Applications in Pattern Recognition*, IEEE Trans. Inform. Theory **21** (1975), no. 1, 32–40.

[HHLS16] M.T. Harandi, R. Hartley, B. Lovell, and C. Sanderson, *Sparse Coding on Symmetric Positive Definite Manifolds Using Bregman Divergences*, IEEE Transactions on Neural Networks and Learning Systems **27** (2016), no. 6, 1294–1306.

[Hig08] N.J. Higham, *Functions of Matrices: Theory and Computation*, SIAM, 2008.

[HP11] S. Har-Peled, *Geometric Approximation Algorithms*, AMS, 2011.

[HSPS19] R. Hühnerbein, F. Savarino, S. Petra, and C. Schnörr, *Learning Adaptive Regularization for Image Labeling Using Geometric Assignment*, Proc. SSVM, Springer, 2019.

[HSS08] T. Hofmann, B. Schölkopf, and A. J. Smola, *Kernel Methods in Machine Learning*, Ann. Statistics **36** (2008), no. 3, 1171–1220.

[Jos17] J. Jost, *Riemannian Geometry and Geometric Analysis*, 7th ed., Springer-Verlag Berlin Heidelberg, 2017.

[KAH+15] J.H. Kappes, B. Andres, F.A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B.X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother, *A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems*, International Journal of Computer Vision **115** (2015), no. 2, 155–184.

[Kar77] H. Karcher, *Riemannian Center of Mass and Mollifier Smoothing*, Comm. Pure Appl. Math. **30** (1977), 509–541.

[KMBB15] Andreas Kleefeld, Anke Meyer-Baese, and Bernhard Burgeth, *Elementary morphology for so (2)-and so (3)-orientation fields*, International Symposium on Mathematical Morphology and Its Applications to Signal and Image Processing, Springer, 2015, pp. 458–469.

[Lee13] J. M. Lee, *Introduction to Smooth Manifolds*, Springer, 2013.

[MP00] G. McLachlan and D. Peel, *Finite Mixture Models*, Wiley, 2000.

[RW09] R. T. Rockafellar and R. J.-B. Wets, *Variational Analysis*, 3rd ed., Springer, 2009.

[SM09] R. Subbarao and P. Meer, *Nonlinear Mean Shift over Riemannian Manifolds*, Int. J. Comp. Vision **84** (2009), no. 1, 1–20.

[Sra13] S. Sra, *Positive Definite Matrices and the Symmetric Stein Divergence*, CoRR abs/1110.1773 (2013).

[Teb07] M. Teboulle, *A Unified Continuous Optimization Framework for Center-Based Clustering Methods*, J. Mach. Learning Res. **8** (2007), 65–102.

[TPM06] O. Tuzel, F. Porikli, and P. Meer, *Region covariance: A fast descriptor for detection and classification*, Proc. ECCV, Springer, 2006, pp. 589–600.

[TS16] P.K. Turaga and A. Srivastava (eds.), *Riemannian Computing in Computer Vision*, Springer, 2016.

[ZSPS18] A. Zeilmann, F. Savarino, S. Petra, and C. Schnörr, *Geometric Numerical Integration of the Assignment Flow*, CoRR abs/1810.06970 (2018).

[ZZr+18] A. Zern, M. Zisler, F. Åström, S. Petra, and C. Schnörr, *Unsupervised Label Learning on Manifolds by Spatially Regularized Geometric Assignment*, Proc. GCPR, 2018.

(A. Zern) IMAGE AND PATTERN ANALYSIS GROUP, HEIDELBERG UNIVERSITY, GERMANY
E-mail address: artjom.zern@iwr.uni-heidelberg.de

(M. Zisler) IMAGE AND PATTERN ANALYSIS GROUP, HEIDELBERG UNIVERSITY, GERMANY
E-mail address: zisler@math.uni-heidelberg.de

(S. Petra) MATHEMATICAL IMAGING GROUP, HEIDELBERG UNIVERSITY, GERMANY
E-mail address: petra@math.uni-heidelberg.de
URL: https://www.stpetra.com

(C. Schnörr) IMAGE AND PATTERN ANALYSIS GROUP, HEIDELBERG UNIVERSITY, GERMANY
E-mail address: schnoerr@math.uni-heidelberg.de
URL: https://ipa.math.uni-heidelberg.de