SELF-ASSIGNMENT FLOWS FOR UNSUPERVISED DATA LABELING ON GRAPHS

MATTHIAS ZISLER, ARTJOM ZERN, STEFANIA PETRA, CHRISTOPH SCHNÖRR

ABSTRACT. This paper extends the recently introduced assignment flow approach for supervised image labeling to unsupervised scenarios where no labels are given. The resulting self-assignment flow takes a pairwise data affinity matrix as input data and maximizes the correlation with a low-rank matrix that is parametrized by the variables of the assignment flow, which entails an assignment of the data to themselves through the formation of latent labels (feature prototypes). A single user parameter, the neighborhood size for the geometric regularization of assignments, drives the entire process. By smooth geodesic interpolation between different normalizations of self-assignment matrices on the positive definite matrix manifold, a one-parameter family of self-assignment flows is defined. Accordingly, our approach can be characterized from different viewpoints, e.g. as performing spatially regularized, rank-constrained discrete optimal transport, or as computing spatially regularized normalized spectral cuts. Regarding combinatorial optimization, our approach successfully determines completely positive factorizations of self-assignments in large-scale scenarios, subject to spatial regularization. Various experiments including the unsupervised learning of patch dictionaries using a locally invariant distance function, illustrate the properties of the approach.

CONTENTS

Key words and phrases. unsupervised learning, regularized clustering, graph partitioning, image labeling, assignment manifold, assignment flow, replicator equation, evolutionary game dynamics, dynamical systems.

1. Introduction

Overview, contribution. Assignment flows [ÅPSS17] correspond to a smooth dynamical system for contextual data labeling (classification) on an arbitrary given graph. The basic *supervised* setting assumes a set of prototypes to be given, that are assigned to the data by numerically computing the flow. ‘Contextual’ means that decisions within local neighborhoods affect each other and are taken into account.

Assignment flows are defined using information geometry [Lau87, AN00]. An elementary statistical manifold provides both a target space for data embedding and a state space on which the assignment flow evolves. Corresponding vector fields are parametrized and thus enable to learn the adaptivity of contextual label assignments, rather than parameters of a fixed regularizer as with traditional graphical models of variational approaches to inverse problems. Modular compositional design facilitates extensions beyond the basic supervised scenario, including those investigated in the present paper. Smoothness enables the design of efficient algorithms using geometric integration [ZSPS19]. The assignment flow for supervised labeling is specified in Section 2.5. We refer to [Sch19] for further discussion and a review of our recent work.

The availability of prototypes as class representatives is a strong requirement in practice. In many applications either prototypes are not available or it is not clear what prototypes represent the classes properly. A basic remedy is to cluster the data in a preprocessing step. However, the clustering step then does not take into account the framework in which the resulting prototypes are subsequently used for classification. In our recent work [ZZPS19a], we took a step towards a more natural approach: the assignment flow for supervised classification was extended so as to enable the *adaption* of prespecified prototypes. While this adaption is based on the *same* framework that is used for subsequent contextual classification, some initial prototypes still have to be given.

In this paper, we adopt a *completely unsupervised* scenario where no prototypes are given at all. Data are merely given in terms of pairwise distances or affinity values forming a distance or affinity matrix. This includes the basic scenarios of pattern recognition and machine learning: distances between Euclidean feature vectors, Riemannian distances between manifold-valued features, and kernel matrices after embedding given feature vectors into reproducing kernel Hilbert space (RKHS) [HSS08]. Our approach utilizes various relaxations of a graph partitioning problem that naturally arises when the missing prototypes of the supervised setting are removed and replaced by a copy of the given data, from which prototypes have to be learned from scratch. The relaxations involve variants of corresponding self-assignment matrices that are parametrized by the assignment flow. A key parameter is the *scale* of the supervised assignment flow in terms of the size of local neighborhoods where evolving assignments driven by the flow affect each other. This parameter determines how fine or coarse the resulting partition is, and how many corresponding prototypes can be recovered under additional assumptions.

A key property of our approach is that *no bias* affects the emergence of these prototypes, and that the *very same* framework is used for *both learning* these prototypes *and* subsequent contextual data *labeling* (classification). In addition, a *single* component of the supervised assignment flow has only to be modified in order to extend this approach to the completely unsupervised setting. In particular, geometric schemes for numerically integrating the assignment flow [ZSPS19] still apply.

Related work. The literature on clustering and unsupervised learning is vast. No attempt is made to review it here. We confine ourselves to elucidating common and different aspects of our approach from three different viewpoints that have become prominent in the literature: (i) spectral relaxation and clustering using normalized graph Laplacians [SM00, vL07]; (ii) regularized transport of discrete probability measures [BCPD99, Pey18]; (iii) matrix factorization and aspects of combinatorial optimization [RW95, ZS05, KYP15, YC16]. From each viewpoint, our approach can be characterized as combining tight relaxation of graph partitioning, geometric spatial regularization of assignments, and geometric numerical integration in a mathematically novel way. The present paper considerably elaborates the conference version [ZZPS19b].

Organization. We introduce basic notation and collect background in Section 2, including the supervised assignment flow as basic framework. Section 3 shows how the graph partitioning problem and various relaxations emerge within this framework, after replacing the prototypes by the data and assigning them to themselves. We highlight differences between two major relaxations and show how latent prototypes emerge as the assignment flow evolves. After terminating the self-assignment flow at some labeling, these prototypes can be recovered explicitly under additional assumptions (weighted averaging in feature space has to be well-defined and computationally feasible). A family of self-assignment flows, based on the relaxations of Section 3, is defined in Section 4. It is shown that the latent prototypes maximize cluster separability. In this sense, the self-assignment flow performs *self-supervision*. Related work is discussed in Section 5. The approach is illustrated in Section 6 using various basic examples of image analysis and more advanced examples, including unsupervised and locally invariant patch learning, assignment, and transfer to novel data. In order to highlight the broad applicability of our approach, an experiment using weighted graph data is included, too.

2. Preliminaries

We collect in this section material required in subsequent sections.

2.1. **Basic Notation.** We set $[n] = \{1, 2, \dots, n\}$ for $n \in \mathbb{N}$ and $\mathbb{1}_n = (1, 1, \dots, 1)^\top \in \mathbb{R}^n$. The cardinality of a finite set S is denoted by $|S|$. The following spaces of matrices will be used.

- $\mathbb{S}^n$: symmetric $n \times n$ matrices
- $\mathbb{S}^n_+$: symmetric nonnegative $n \times n$ matrices
- $\mathbb{R}^{n\times c}_+$: nonnegative $n \times c$ matrices
- $\mathcal{P}^c$: symmetric positive definite $c \times c$ matrices

$\|\cdot\|$ denotes the Euclidean norm and the Frobenius norm for vectors and matrices, respectively. All other norms will be indicated by a corresponding subscript. For a matrix $A \in \mathbb{R}^{n\times c}$, A_i, $i \in [n]$ denote the row vectors and A^j, $j \in [c]$ denote the column vectors, $A^\top \in \mathbb{R}^{c\times n}$ the transpose and $A^\dagger$ the Moore-Penrose generalized inverse of A. $\operatorname{tr}(A) = \sum_{i\in[n]} A_{i,i}$ denotes the trace of a square matrix $A \in \mathbb{R}^{n\times n}$. $\Delta_n = \{p \in \mathbb{R}^n_+ \colon \langle \mathbb{1}_n, p\rangle = 1\}$ denotes the probability simplex. The orthogonal projection onto a closed convex set C is denoted by Π_C.

2.2. **Scatter Matrices.** We collect basic concepts of statistical pattern recognition [DK82]. They will be used for interpreting self-assignment flows from a corresponding angle in Section 4.3.

Let

$$\mathcal{F}_n = \{f_i \in \mathbb{R}^d,\ i \in \mathcal{I}\} \tag{2.1}$$

denote given data in terms of feature vectors in a Euclidean space. Suppose these data are classified corresponding to the partition

$$\mathcal{I} = \dot{\bigcup}_{j\in[c]} \mathcal{I}_j, \qquad |\mathcal{I}_j| = n_j, \qquad \sum_{j\in[c]} n_j = n = |\mathcal{I}|, \qquad c \in \mathbb{N}, \tag{2.2}$$

that is, datum f_i belongs to class j iff $i \in \mathcal{I}_j$.

We define the empirical quantities

$$P_j = \frac{n_j}{n}, \qquad j \in [c] \qquad\qquad \text{(prior probabilities)} \tag{2.3a}$$

$$m_j = \frac{1}{n_j} \sum_{i\in\mathcal{I}_j} f_i, \qquad j \in [c] \qquad\qquad \text{(class-conditional mean vectors)} \tag{2.3b}$$

$$m = \sum_{j\in[c]} P_j m_j = \frac{1}{n} \sum_{i\in[n]} f_i \qquad\qquad \text{(mean vector)} \tag{2.3c}$$

and the *scatter matrices* (empirical covariance matrices)

$$S_t = \frac{1}{n} \sum_{i\in[n]} (f_i - m)(f_i - m)^\top, \tag{2.4a}$$

$$S_w = \sum_{j\in[c]} P_j \cdot \frac{1}{n_j} \sum_{i\in\mathcal{I}_j} (f_i - m_j)(f_i - m_j)^\top = \frac{1}{n} \sum_{j\in[c]} \sum_{i\in\mathcal{I}_j} (f_i - m_j)(f_i - m_j)^\top, \tag{2.4b}$$

$$S_b = \sum_{j\in[c]} P_j (m_j - m)(m_j - m)^\top. \tag{2.4c}$$

S_w is called the *within-class* scatter matrix, whereas S_b is called the *between-class* scatter matrix. S_t is called the *total* scatter matrix since

$$S_t = S_w + S_b, \tag{2.5}$$

as an elementary computation shows.

In *supervised* scenarios the class-label assignments $i \in \mathcal{I}_j$ are known and the decomposition (2.5) can be computed. Assuming S_w has full rank, a basic objective for dimension reduction by extracting lower-dimensional features from the data $\mathcal{F}_n$ is then given by the class-separability measure

$$\operatorname{tr}(S_w^{-1} S_b). \tag{2.6}$$

Defining the features by $Y^\top x$, for some matrix $Y \in \mathbb{R}^{d \times c}$ to be determined, transforms (2.6) to $\operatorname{tr}((Y^\top S_w Y)^{-1} Y^\top S_b Y)$. Maximizing this objective with respect to Y *simultaneously* maximizes the between-class variation and minimizes the within-class variation. The column vectors of the optimal Y are given by dominant generalized eigenvectors of the matrix pencil (S_b, S_w). The map $Y^\top x$ to a lower-dimensional space preserves the structure of the data, as represented by the scatter matrices S_w, S_b, as much as possible.

Our viewpoint in this paper differs. Since we assume unlabelled (unclassified) data, the decomposition (2.5) is unknown. Accordingly, we are interested in the more involved problem to compute class representatives m_j, $j \in [c]$ from the data $\mathcal{F}_n$ so as to obtain good clusters based on the objective (2.6), see Section 4.3.

2.3. **Sketching Large Affinity Matrices.** In order to cope with large-scale scenarios, we will have to compress large symmetric and positive semi-definite matrices $K \in \mathbb{S}^n$. The problem is to obtain a computationally feasible approximation of the best rank-ℓ approximation

$$K_\ell = U_1 D_\ell(K) U_1^\top, \qquad \ell \ll n, \tag{2.7}$$

where D_ℓ and $U_1 \in \mathbb{R}^{n \times \ell}$ contain the dominant eigenvalues and eigenvectors of the spectral decomposition $K = UD(K)U^\top$. Computing (2.7) directly for large n using the Singular Value Decomposition (SVD) is too expensive. Computationally feasible approximations [GM16] result in the *compressed matrix*

$$\widehat{K}_\ell = CA^\dagger C^\top \tag{2.8a}$$

that is parametrized by a *sketching matrix* $S \in \mathbb{R}^{n \times \ell}$ with

$$C = K^q S, \qquad A = S^\top K^{2q-1} S, \qquad q \in \mathbb{N} \tag{2.8b}$$

and hence has rank at most ℓ. $A^\dagger$ is the Moore-Penrose generalized inverse of A and $q \in \{1,2,3\}$ is a small integer in practice. Choosing $q > 1$ is more expensive due to the multiplication of the large matrix K of (2.8b) but yields in theory a better approximation of (2.7) by (2.8a) with respect to the spectral norm.

In this paper, we confine ourselves to the following computationally cheap version of this method for computing (2.8a), based on *uniform sampling* of ℓ columns directly from K. Assuming w.l.o.g. that they form the first ℓ columns of K, the corresponding partition $[n] = [\ell] \cup \big([n] \setminus [\ell]\big)$ and $S = \big(\begin{smallmatrix} I_\ell \\ 0 \end{smallmatrix}\big)$ yields with $q = 1$

$$K = \begin{pmatrix} A & B_1 \\ B_1 & B_2 \end{pmatrix}, \qquad C = \begin{pmatrix} A \\ B_1 \end{pmatrix}, \tag{2.9}$$

and using $AA^\dagger A = A$,

$$\widehat{K}_\ell = \begin{pmatrix} A \\ B_1 \end{pmatrix} A^\dagger \begin{pmatrix} A & B_1 \end{pmatrix} = \begin{pmatrix} A & AA^\dagger B_1 \\ B_1 A^\dagger A & B_1 A^\dagger B_1 \end{pmatrix}. \tag{2.10}$$

Assuming the A has full rank, we obtain the classical *Nyström extension*

$$\widehat{K}_\ell = \begin{pmatrix} A & B_1 \\ B_1 & B_1 A^{-1} B_1 \end{pmatrix} \tag{2.11}$$

introduced in machine learning by [WS01], studied much earlier in linear algebra – see, e.g., the Schur compression matrix and references in [And79] – and analyzed by [DM05].

2.4. **The Positive Definite Matrix Manifold $\mathcal{P}^n$.** The following is taken from [Bha06]. The set

$$\mathcal{P}^n = \{S \in \mathbb{S}^n \colon \lambda_i(S) > 0,\ \forall i \in [n]\} \tag{2.12}$$

of symmetric and positive definite matrices form a smooth Riemannian manifold with tangent spaces $T_S\mathcal{P}^n \cong \mathbb{S}^n$ identified with $\mathbb{S}^n$ and Riemannian metric

$$\langle S_1, S_2\rangle_S = \operatorname{tr}(S^{-1}S_1S^{-1}S_2), \qquad S_1, S_2 \in \mathbb{S}^n, \quad S \in \mathcal{P}^n \tag{2.13a}$$

and corresponding norm

$$\|T\|_S = \|S^{-1/2}TS^{-1/2}\|, \qquad T \in \mathbb{S}^n, \quad S \in \mathcal{P}^n. \tag{2.13b}$$

For any $A, B \in \mathcal{P}^n$, there exists a unique geodesic joining A and B given by

$$\gamma(s) = A^{1/2}\big(A^{-1/2}BA^{-1/2}\big)^s A^{1/2}, \qquad s \in [0,1]. \tag{2.14}$$

2.5. **Representation of Assignments.** The assignment flow is a basic dynamical system for labeling data given on a graph [ÅPSS17]. We refer to [Sch19] for the mathematical background and a review of recent developments.

2.5.1. *Assignment Manifold.* Let $(\mathcal{F}, d_{\mathcal{F}})$ be a metric space and

$$\mathcal{F}_n = \{f_i \in \mathcal{F} \colon i \in \mathcal{I}\}, \qquad |\mathcal{I}| = n. \tag{2.15}$$

given data. Assume that a predefined set of **prototypes**

$$\mathcal{F}_* = \{f^*_j \in \mathcal{F} \colon j \in \mathcal{J}\}, \qquad |\mathcal{J}| = c. \tag{2.16}$$

is given. *Data labeling* denotes the assignments

$$j \to i, \qquad f^*_j \to f_i \tag{2.17}$$

of a single prototype $f^*_j \in \mathcal{F}_*$ to each data point $f_i \in \mathcal{F}_n$. The set $\mathcal{I}$ is assumed to form the vertex set of an undirected graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$ which defines a relation $\mathcal{E} \subset \mathcal{I} \times \mathcal{I}$ and neighborhoods

$$\mathcal{N}_i = \{k \in \mathcal{I} \colon ik \in \mathcal{E}\} \cup \{i\}, \tag{2.18}$$

where ik is a shorthand for the unordered pair (edge) $(i,k) = (k,i)$.

The assignments (labeling) (2.17) are represented by matrices in the set

$$\mathcal{W}^c_* = \big\{W \in \{0,1\}^{n\times c} \colon W\mathbb{1}_c = \mathbb{1}_n,\ \operatorname{rank}(W) = c\big\} \tag{2.19}$$

with unit vectors W_i, $i \in \mathcal{I}$, called **assignment vectors**, as row vectors. Moreover the rank constraint ensures that exactly c labels are assigned. These assignment vectors are computed by numerically integrating the assignment flow below (2.36), in the following elementary geometric setting. The integrality constraint and the rank constraint of (2.19) is relaxed and vectors

$$W_i = (W_{i,1}, \dots, W_{i,c})^\top \in \mathcal{S}, \quad i \in \mathcal{I}, \tag{2.20}$$

that we still call *assignment vectors*, are considered on the elementary Riemannian manifold

$$(\mathcal{S}, g), \qquad \mathcal{S} = \{p \in \Delta_c \colon p > 0\} \tag{2.21}$$

with

$$\mathbb{1}_{\mathcal{S}} = \frac{1}{c}\mathbb{1} \in \mathcal{S}, \qquad (\textbf{barycenter}) \tag{2.22}$$

tangent space

$$T_0 = \{v \in \mathbb{R}^c \colon \langle \mathbb{1}, v\rangle = 0\} \tag{2.23}$$

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

$$\Pi_0 \colon \mathbb{R}^c \to T_0, \qquad \Pi_0 = \Pi_{T_0} = I - \mathbb{1}_{\mathcal{S}}\mathbb{1}^\top \tag{2.24}$$

and the Fisher-Rao metric

$$g_p(u,v) = \sum_{j\in\mathcal{J}} \frac{u^j v^j}{p^j}, \quad p \in \mathcal{S}, \quad u,v \in T_0. \tag{2.25}$$

Based on the linear map

$$R_p \colon \mathbb{R}^c \to T_0, \qquad R_p = \operatorname{Diag}(p) - pp^\top, \qquad p \in \mathcal{S} \tag{2.26}$$

satisfying

$$R_p = R_p\Pi_0 = \Pi_0 R_p, \tag{2.27}$$

exponential maps and their inverses are defined as

$$\begin{aligned}
&\operatorname{Exp}\colon \mathcal{S}\times T_0 \to \mathcal{S}, && (p,v) \mapsto \operatorname{Exp}_p(v) = \frac{pe^{\frac{v}{p}}}{\langle p, e^{\frac{v}{p}}\rangle}, && \text{(2.28a)}\\
&\operatorname{Exp}_p^{-1}\colon \mathcal{S} \to T_0, && q \mapsto \operatorname{Exp}_p^{-1}(q) = R_p \log\frac{q}{p}, && \text{(2.28b)}\\
&\exp_p\colon T_0 \to \mathcal{S}, && \exp_p = \operatorname{Exp}_p \circ R_p, && \text{(2.28c)}\\
&\exp_p^{-1}\colon \mathcal{S} \to T_0, && \exp_p^{-1}(q) = \Pi_0 \log\frac{q}{p}. && \text{(2.28d)}
\end{aligned}$$

Remark 2.1. Applying the map $\exp_p$ to a vector in $\mathbb{R}^c = T_0 \oplus \mathbb{R}\mathbb{1}$ does not depend on the constant component of the argument, due to (2.27).

Remark 2.2. The map Exp corresponds to the e-connection of information geometry, rather than to the exponential map of the Riemannian connection [AN00]. Accordingly, the affine geodesics (2.28a) are not length-minimizing. But they provide a close approximation and are more convenient for numerical computations.

The **assignment manifold** is defined as

$$(\mathcal{W}, g), \qquad \mathcal{W} = \mathcal{S}\times\cdots\times\mathcal{S}. \qquad (n = |\mathcal{I}| \text{ factors}) \tag{2.29}$$

Points $W \in \mathcal{W}$ are row-stochastic matrices $W \in \mathbb{R}^{n\times c}$ with row vectors $W_i \in \mathcal{S},\ i \in \mathcal{I}$ that represent the assignments (2.17) for every $i \in \mathcal{I}$. We set

$$\mathcal{T}_0 = T_0 \times\cdots\times T_0 \qquad (n = |\mathcal{I}| \text{ factors}) \tag{2.30}$$

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

2.5.2. *Assignment Flow.* Based on (2.15) and (2.16), the distance vector field

$$D_{\mathcal{F};i} = \big(d_{\mathcal{F}}(f_i, f_1^*), \dots, d_{\mathcal{F}}(f_i, f_c^*)\big)^\top, \qquad i \in \mathcal{I} \tag{2.31}$$

is well-defined. These vectors are collected as row vectors of the **distance matrix**

$$D_{\mathcal{F}} \in \mathbb{R}^{n\times c}_+. \tag{2.32}$$

The **likelihood map** and the **likelihood vectors**, respectively, are defined as

$$L_i\colon \mathcal{S}\to\mathcal{S}, \qquad L_i(W_i) = \exp_{W_i}\Big(-\frac{1}{\rho}D_{\mathcal{F};i}\Big) = \frac{W_i e^{-\frac{1}{\rho}D_{\mathcal{F};i}}}{\langle W_i, e^{-\frac{1}{\rho}D_{\mathcal{F};i}}\rangle}, \qquad i \in \mathcal{I}, \tag{2.33}$$

where the scaling parameter $\rho > 0$ is used for normalizing the a-priori unknown scale of the components of $D_{\mathcal{F};i}$ that depends on the specific application at hand.

A key component of the assignment flow is the interaction of the likelihood vectors through *geometric* averaging within the local neighborhoods (2.18). Specifically, using the weights

$$\Omega_i = \Big\{ w_{i,k} \colon k \in \mathcal{N}_i,\ w_{i,k} > 0,\ \sum_{k \in \mathcal{N}_i} w_{i,k} = 1 \Big\}, \quad i \in \mathcal{I}, \tag{2.34}$$

the **similarity map** and the **similarity vectors**, respectively, are defined as

$$S_i \colon \mathcal{W} \to \mathcal{S}, \qquad S_i(W) = \mathrm{Exp}_{W_i}\Big(\sum_{k \in \mathcal{N}_i} w_{i,k} \, \mathrm{Exp}_{W_i}^{-1}\big(L_k(W_k)\big)\Big), \qquad i \in \mathcal{I}. \tag{2.35}$$

If Exp_{W_i} were the exponential map of the Riemannian (Levi-Civita) connection, then the argument inside the brackets of the right-hand side would just be the negative Riemannian gradient with respect to W_i of the center of mass objective function comprising the points L_k, $k \in \mathcal{N}_i$, i.e. the weighted sum of the squared Riemannian distances between W_i and L_k [Jos17, Lemma 6.9.4]. In view of Remark 2.2, this interpretation is only approximately true mathematically, but still correct informally: $S_i(W)$ moves W_i towards the geometric mean of the likelihood vectors L_k, $k \in \mathcal{N}_i$. Since $\mathrm{Exp}_{W_i}(0) = W_i$, this mean is equal to W_i if the aforementioned gradient vanishes.

The **assignment flow** is induced by the system of nonlinear ODEs

$$\dot{W} = R_W S(W), \qquad W(0) = \mathbb{1}_{\mathcal{W}}, \tag{2.36a}$$

$$\dot{W}_i = R_{W_i} S_i(W), \qquad W_i(0) = \mathbb{1}_{\mathcal{S}}, \quad i \in \mathcal{I}, \tag{2.36b}$$

where $\mathbb{1}_{\mathcal{W}} \in \mathcal{W}$ denotes the barycenter of the assignment manifold (2.29). System (2.36a) collects all systems (2.36b), for every vertex $i \in \mathcal{I}$. The latter systems are coupled within local neighborhoods $\mathcal{N}_i$ due to the similarity vectors $S_i(W)$ given by (2.35). The solution $W(t) \in \mathcal{W}$ is numerically computed by geometric integration [ZSPS19] and determines a labeling $W(T) \in \mathcal{W}_*^c$ for sufficiently large T after a trivial rounding operation.

2.6. **Greedy k-Center Metric Clustering.** In order to handle large-scale scenarios, the following simple but effective algorithm from [HP11] can be employed for data reduction in a preprocessing step. The algorithm approximates the k-center clustering along with a *performance guarantee* and only requires *linear complexity* $\mathcal{O}(nc)$ with respect to the (large) number of data points n. By using a min-max objective (see (2.38) below), selected data points are evenly spread among all data points and hence do not introduce a bias beforehand.

The task of k-center clustering is as follows. Given data points $\mathcal{F}_n$ from a metric space $(\mathcal{F}, d_{\mathcal{F}})$, determine a subset

$$\mathcal{F}_c = \{f_j \colon j \in \mathcal{J}\} \subset \mathcal{F}_n, \qquad |\mathcal{J}| = c. \tag{2.37}$$

that solves the combinatorially hard optimization problem

$$E_\infty^* = \min_{\mathcal{F}_c \subset \mathcal{F}_n, |\mathcal{F}_c| = c} E_\infty(\mathcal{F}_c), \qquad E_\infty(\mathcal{F}_c) = \max_{f \in \mathcal{F}_n} d_{\mathcal{F}}(f, \mathcal{F}_c), \tag{2.38}$$

where $d_{\mathcal{F}}(f, \mathcal{F}_c) = \min_{f' \in \mathcal{F}_c} d_{\mathcal{F}}(f, f')$.

A greedy approximation is computed as follows. Start with a first initial point f_1, e.g. chosen randomly in $\mathcal{F}_n$. Then select the remaining $c-1$ points $f_2, \dots, f_c$ successively by determining the point that is most distant from the current subset of already selected points, to obtain a set $\mathcal{F}_c$ that is a 2-approximation $E_\infty(\mathcal{F}_c) \le 2E_\infty^*$ of the optimum (2.38) [HP11, Thm. 4.3]. As a consequence, the subset of c points of $\mathcal{F}_c$ are almost uniformly distributed within $\mathcal{F}_n$, as measured by the metric $d_{\mathcal{F}}$.

3. Self-Assignment

This section prepares the generalisation of the assignment flow (2.36) from supervised labeling to completely unsupervised labeling, where prototypes (2.16) no longer are involved but are determined simultaneously. Our approach is (i) to assign given data (2.15) to itself in terms of a self-affinity matrix parameterized by the assignment matrix $W \in \mathbb{R}^{n\times c}$ (Section 3.2), which ensures computational feasibility since $c \ll n$, and (ii) to generalize later on the likelihood map (2.33) accordingly (Section 4.1). Except for this more general definition of the likelihood map $L(W)$, the subsequent similarity map $S(W)$ given by (2.35) remains unaltered, as do numerical schemes for integrating the flow (2.36) [ZSPS19].

In fact, we define a one-parameter family of self-assignment matrices by geodesic interpolation of two extreme points on the positive definite manifold, that admit natural probabilistic interpretations of the corresponding self-assignments. Properties of these matrices also provide the basis for the interpretation of the resulting self-assignment flows (Section 3.3) and for pointing out connections to related work (Section 5).

3.1. **From Labeling to Partitioning.** Since the prototypes $\mathcal{F}_*$ are unknown, we replace them by the given data $\mathcal{F}_n$. Along with $\mathcal{F}_n$ and the underlying graph $\mathcal{G} = (\mathcal{I}, \mathcal{E})$, we assume a weighted similarity matrix

$$K_{\mathcal{F}} \in \mathbb{S}^n_+ \tag{3.1}$$

to be given with entries

$$K_{\mathcal{F};i,k} = (K_{\mathcal{F}})_{i,k} = k_{\mathcal{F}}(f_i, f_k), \qquad i,k \in \mathcal{I} \tag{3.2}$$

measuring the similarity of the data points f_i, f_k in terms of a nonnegative symmetric function $k_{\mathcal{F}}$. Matrix $K_{\mathcal{F}}$ is positive definite if $k_{\mathcal{F}}$ evaluates the inner product of a data embedding into a corresponding reproducing kernel Hilbert space (RKHS) space [HSS08]. A basic example is a Euclidean feature space $(\mathcal{F}, d_{\mathcal{F}})$ with norm $d_{\mathcal{F}}(f_i, f_k) = \|f_i - f_k\|$ and

$$k_{\mathcal{F}}(f_i, f_k) = e^{-d_{\mathcal{F}}(f_i,f_k)^2/\sigma^2}. \tag{3.3}$$

Let $W \in \mathcal{W}^c_*$ be a labeling. The column vectors W^j, $j \in \mathcal{J}$, of W indicate which data points f_i are assigned to j-th cluster $\mathcal{I}_j$ corresponding to the partition

$$\mathcal{I} = \dot{\bigcup_{j\in\mathcal{J}}} \mathcal{I}_j \tag{3.4}$$

of the data set $\mathcal{F}_n$. Define the diagonal matrix

$$C(W) := \operatorname{Diag}(W^\top \mathbb{1}_n) = \operatorname{Diag}(n_1, \dots, n_c) \in \mathbb{S}^c_+ \tag{3.5a}$$

with the cardinalities of each cluster

$$n_j := |\mathcal{I}_j|, \qquad j \in \mathcal{J} \tag{3.5b}$$

as entries. The quadratic form

$$\frac{1}{2}\langle W^j, K_{\mathcal{F}} W^j\rangle = \frac{1}{2}\sum_{i,k\in\mathcal{I}} k_{\mathcal{F}}(f_i, f_k) W_{i,j} W_{k,j} = \frac{1}{2}\sum_{i\in\mathcal{I}_j} k_{\mathcal{F}}(f_i, f_i) + \sum_{i,k\in\mathcal{I}_j :\, i\neq k} k_{\mathcal{F}}(f_i, f_k) \tag{3.6}$$

measures the *size* of cluster $\mathcal{I}_j$ by the first sum of the right-hand side, which for common kernel functions like (3.3) is proportional to the number n_j of data points assigned to cluster j, and the *connectivity* in terms of the weights $k_{\mathcal{F}}(f_i, f_k)$ of all edges $ik \in \mathcal{E}$ connecting points i and k in this cluster. Assuming that all clusters are non-empty, which amounts to the assumption

$$\operatorname{rank}(W) = c, \tag{3.7}$$

we normalize the preceding expression by the cardinality and sum over all clusters, to obtain

$$\begin{aligned}\sum_{j\in\mathcal{J}}\frac{1}{2n_j}\langle W^j, K_{\mathcal{F}}W^j\rangle &= \frac{1}{2}\sum_{j\in\mathcal{J}}\frac{1}{n_j}\sum_{i\in\mathcal{I}_j}k_{\mathcal{F}}(f_i,f_i)+\sum_{j\in\mathcal{J}}\frac{1}{n_j}\sum_{i,k\in\mathcal{I}_j\colon i\neq k}k_{\mathcal{F}}(f_i,f_k) && (3.8a)\\ &= \frac{1}{2}\sum_{j\in\mathcal{J}}\frac{1}{n_j}(W^{\top}K_{\mathcal{F}}W)_{j,j}\overset{(3.5a)}{=}\frac{1}{2}\operatorname{tr}\big(C(W)^{-1}W^{\top}K_{\mathcal{F}}W\big) && (3.8b)\\ &= \frac{1}{2}\operatorname{tr}\big(K_{\mathcal{F}}A_0(W)\big), && (3.8c)\end{aligned}$$

with

$$A_0(W)=WC(W)^{-1}W^{\top},\qquad W\in\mathcal{W}_*^c. \tag{3.8d}$$

For common kernel functions like (3.3), the first sum of the right-hand side of (3.8a) is just a constant. Objective (3.8c) therefore essentially measures the normalized similarity weights *not* cut by the partition of the underlying graph.

Thus, the problem to partition the data and the underlying graph into c clusters takes the form

$$\max_{W}\operatorname{tr}\big(K_{\mathcal{F}}A_0(W)\big)\qquad\text{subject to}\qquad W\in\mathcal{W}_*^c. \tag{3.9}$$

We record basic properties of the matrix $A_0(W)$.

Lemma 3.1. *Let $W\in\mathcal{W}_*^c$. Then the matrix $A_0(W)$ given by (3.8d) is*

(a) nonnegative and symmetric,

(b) doubly stochastic,

$$A_0(W)\mathbb{1}_n=A_0(W)^{\top}\mathbb{1}_n=\mathbb{1}_n, \tag{3.10}$$

(c) and completely positive,

$$A_0(W)=YY^{\top},\qquad Y\geq 0. \tag{3.11}$$

Proof. (a) is immediate. (b) follows from (3.5a) and the constraint $W\in\mathcal{W}_*^c$ (recall (2.19)). (c) holds with $Y=Y(W)=WC(W)^{-1/2}$. □

Property (c) reflects the combinatorial difficulty of problem (3.9) – see, e.g., [Bom18] and references therein. Therefore, we next discuss various relaxations.

3.2. **Self-Assignment Matrices, Relaxation.**

We start with the definitions of two basic self-assignment matrices. The first relaxation based on (3.8d) drops both the integrality constraint and the rank constraint.

Definition 3.1 (Self-Affinity Matrix). The *self-affinity matrix* is defined as the factorization

$$A_0(W):=WC(W)^{-1}W^{\top},\qquad W\in\mathcal{W}. \tag{3.12}$$

The second definition is based on the observation that equivalent expressions for the normalizing matrix

$$C(W)=W^{\top}W\qquad\text{if}\qquad W\in\mathcal{W}_*^c \tag{3.13}$$

differ after relaxing the feasible set $\mathcal{W}_*^c$. Dropping the integrality constraint but keeping the rank constraint yields the set of full-rank assignment matrices

$$\mathcal{W}^c=\big\{W\in\mathcal{W}\colon\ \operatorname{rank}(W)=c\big\}\qquad\qquad(\textbf{full-rank assignments}) \tag{3.14}$$

and the following definition.

Definition 3.2 (**Self-Influence Matrix**)**.** The *self-influence matrix* is defined as the factorization

$$A_1(W) := W(W^\top W)^{-1}W^\top, \qquad W \in \mathcal{W}^c. \tag{3.15}$$

Definitions 3.1 and 3.2 differ by the normalizing matrices $C(W)$ and $W^\top W$, both of which are positive definite. It is then natural to define a one-parameter family of factorized matrices in terms of a geodesic (2.14) on the positive definite manifold $\mathcal{P}^c$ that connects $C(W)$ and $W^\top W$, which gives rise to the following definition.

Definition 3.3 (**Self-Assignment Matrix**)**.** The *self-assignment matrix* with parameter s is defined as the factorization

$$A_s(W) := W\gamma_s(W)^{-1}W^\top, \qquad s \in [0,1], \qquad W \in \begin{cases} \mathcal{W}, & \text{if } s = 0, \\ \mathcal{W}^c, & \text{if } s > 0, \end{cases} \tag{3.16a}$$

with normalizing matrix

$$\gamma_s(W) = C(W)^{\frac{1}{2}}\big(C(W)^{-\frac{1}{2}}W^\top W C(W)^{-\frac{1}{2}}\big)^s C(W)^{\frac{1}{2}} \;\in\; \mathcal{P}^c. \tag{3.16b}$$

Note that Definition 3.3 corresponds to Definition 3.1 and 3.2 if $s = 0$ and $s = 1$, respectively.

The following proposition collects properties of self-assignment matrices defined above. Property (h) refers to a relation between matrices $A_1\big(W(t)\big)$ and $A_1\big(W(t')\big)$, for any $t, t' \in [0, T]$: they share the same eigenvalues.

Proposition 3.2 (Properties of Self-Assignment Matrices)**.** *Let $A_0(W)$ and $A_1(W)$ be given Definition 3.1 and 3.2, respectively. Then the following properties either hold or not:*

		self-affinity $A_0(W)$	***self-influence*** $A_1(W)$
	admissible assignments	$W \in \mathcal{W}$	$W \in \mathcal{W}^c$
(a)	*symmetric*	✓	✓
(b)	*positive semi-definite*	✓	✓
(c)	*nonnegative*	✓	✗
(d)	*doubly stochastic*	✓	✗
(e)	*completely positive*	✓	✗
(f)	*rank*	$\leq c$	$= c$
(g)	*orthogonal projection*	✗	$\Pi_{\mathcal{R}(W)}$
(h)	*iso-spectral*	✗	✓
(i)	*eigenvalues* $\in$	$[0,1]$	$\{0,1\}$
(j)	*multiplicity* $(\lambda = 1)$	$= 1$	$= c$
(k)	*multiplicity* $(\lambda = 0)$	$\geq n - c$	$= n - c$
(l)	*eigenvector(s)* $(\lambda = 1)$	$\mathbb{1}_n$	$\big(W(W^\top W)^{-\frac{1}{2}}\big)^j, \quad j \in \mathcal{J}$

Proof. (a)-(f) are clear. We focus on (g)-(l).

(g) On easily checkes that $A_1(W) = A_1(W)^2$ is idempotent whereas $A_0(W)$ is not. Taking into account (a) implies the assertion for $s = 1$.

(h) Follows from (i) and (j) for $s = 1$.

(i) Case $s = 0$. The lower eigenvalue bound 0 follows from (a),(b), the upper bound 1 from (d) and [BP94, Thm. 5.3]. Case $s = 1$. This is immediate due to (g).

(j) Case $s = 0$. $W \in \mathcal{W}$ implies that $A_0(W)$ is strictly positive. (i) and [BP94, Thm. 1.4] then imply the assertion. Case $s = 1$. This is immediate due to (f),(g).

(k) Both assertions follow from (f).
(l) Case $s = 0$ follows from (d) and [BP94, Thm. 5.3]. Case $s = 1$. Setting $Y = W(W^{\top}W)^{-1/2}$, one directly computes $A_1(W)Y = Y$ and $Y^{\top}Y = I_c$.

□

The last definition of this section concerns the 'difference' between the normalizing matrices $C(W)$ and $W^{\top}W$ of Definitions 3.1–3.3.

Definition 3.4 (**Cluster-Confusion Matrix**)**.** The *cluster-confusion matrix* is defined as the matrix factorization

$$B(W) := C(W)^{-1}W^{\top}W \in \mathbb{R}^{c\times c}_{+}, \qquad W \in \mathcal{W}. \tag{3.17}$$

Proposition 3.3 (Properties of the Cluster-Confusion Matrix)**.** *The cluster-confusion matrix $B(W)$ has the following properties:*

(a) *entry-wise positive:* $B(W) > 0$
(b) *row stochastic:* $B(W)\mathbb{1}_c = \mathbb{1}_c$
(c) *pure clusters:* $B(W) = I_c$ *if and only if* $W \in \mathcal{W}^c_*$.
(d) *rank lower bound:* $0 \leq \operatorname{tr}\big(B(W)\big) \leq \operatorname{rank}(W)$ *with equality if* $W \in \mathcal{W}^c_*$

Proof. (a)-(c) directly follow from the definitions of $B(W)$ and W^c_*. (d) follows from $\operatorname{tr}(B(W)) = \operatorname{tr}(A_0(W))$ together with Prop. 3.2 (c) and (i). □

3.3. **Relaxations: Interpretation.**

We take a closer look at the relaxations of problem (3.9).

3.3.1. *Self-Affinity Matrix.* Following [ÅPSS17], we interpret each entry of the assignment matrix $W \in \mathcal{W}$ as posterior probability

$$P(j|i) = W_{i,j}, \qquad j \in \mathcal{J}, \quad i \in \mathcal{I} \tag{3.18}$$

of label j, conditioned on the observation of the data point f_i. According to the completely unsupervised scenario here, we adopt the uniform prior distribution

$$P(i) = \frac{1}{n}, \quad i \in \mathcal{I} \tag{3.19}$$

of the data. Marginalization yields the label distribution

$$P(j) = \sum_{i\in\mathcal{I}} P(j|i)P(i) = \frac{1}{n}\big(W^{\top}\mathbb{1}_n\big)_j, \tag{3.20}$$

which measures the size of cluster $\mathcal{I}_j$ in terms of the relative mass of assignments. Invoking Bayes' rule, we compute the distribution analogous to (3.18), but with the roles of data and labels reversed, to obtain

$$Q(k|j) = \frac{P(j|k)P(k)}{P(j)} = \frac{W_{k,j}}{\sum_{i\in\mathcal{I}} W_{i,j}} = \big(C(W)^{-1}W^{\top}\big)_{j,k}. \tag{3.21}$$

The probability of the self-assignments $f_i \leftrightarrow f_k,\ i,k \in \mathcal{I}$ then result from marginalization over the labels

$$A_{0;i,k}(W) := \sum_{j\in\mathcal{J}} Q(k|j)P(j|i) = \sum_{j\in\mathcal{J}} W_{i,j}\big(C(W)^{-1}W^{\top}\big)_{j,k} = \big(WC(W)^{-1}W^{\top}\big)_{i,k}. \tag{3.22}$$

This expression explains the relaxation that is at the basis of Definition 3.1. It specifies the probability that two vertices i and k get assigned the same label (no matter which one), i.e. that they belong to the same cluster.

Finally, the derivation of problem (3.9) – cf. (3.8) – showed that optimizing the assignments in order to maximize the correlation (inner product) of $A_0(W)$ and $K_{\mathcal{F}}$ amounts to cover the most similar data points by the components of the partition (clusters).

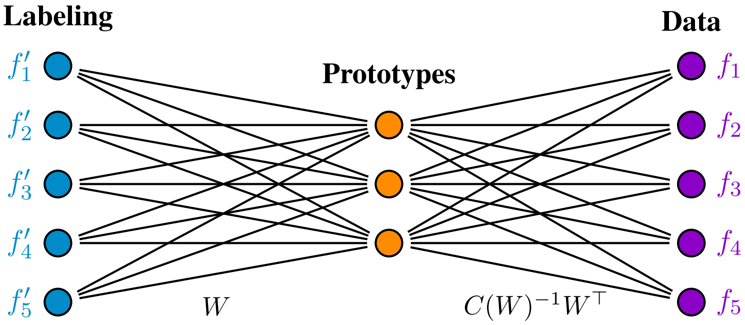


FIGURE 3.1. The self-affinity matrix $A_0(W)$ due to Definition 3.1 comprises the probabilities for each pair of data points $f_i, f_k \in \mathcal{I}$ to belong to the same cluster. The factorization (3.22) of $A_0(W)$ admits the interpretation that optimizing the assignments W implicitly forms prototypes f^*_j, $j \in \mathcal{J}$ that are assigned to the data themselves so as to maximize the correlation with pairwise similarities given as entries of the matrix $K_{\mathcal{F}}$.

3.3.2. *Recovery of Latent Prototypes.* Although problem (3.9) does not involve prototypes (2.16), such prototypes can be recovered from the solution W to the problem relaxation discussed in Section 3.3.1. Specifically, the probabilities $Q(i|j)$ given by (3.21) indicate the contribution of each data point f_i to cluster j. Consequently, adopting the manifold assumption that the data $\mathcal{F}_n$ are sampled on a Riemannian manifold, prototypes can be recovered as weighted Riemannian means by solving

$$f^*_j = \arg\min_{f \in \mathcal{F}} \sum_{i \in \mathcal{I}} \big(C(W)^{-1}W^{\top}\big)_{j,i} d^2_{\mathcal{F}}(f, f_i), \qquad j \in \mathcal{J}. \tag{3.23}$$

In the basic case of Euclidean data $\mathcal{F}_n \subset \mathbb{R}^d$, this problem yields the closed form averages

$$f^*_j = \sum_{i \in \mathcal{I}} \big(C(W)^{-1}W^{\top}\big)_{j,i} f_i, \qquad j \in \mathcal{J}. \tag{3.24}$$

Figure 3.1 illustrates the data self-assignment via the self-affinity matrix and latent prototypes.

3.3.3. *Self-Influence Matrix.* Let $W \in \mathcal{W}^c$ be given and temporarily assume that d-dimensional Euclidean feature vectors are given as data $\mathcal{F}_n$ and collected as row vectors in the matrix

$$F = (f_1, \dots, f_n)^{\top} \in \mathbb{R}^{n \times d}. \tag{3.25}$$

Let the matrix

$$F^* = (f^*_1, \dots, f^*_c)^{\top} \in \mathbb{R}^{c \times d} \tag{3.26}$$

collect the prototypes. Given W and F, a least-squares fit yields

$$F^* = \arg\min_{G \in \mathbb{R}^{c \times d}} \frac{1}{2}\|WG - F\|^2_F = (W^{\top}W)^{-1}W^{\top}F, \tag{3.27}$$

which is well-defined since $W \in \mathcal{W}^c$ has full rank. Using these prototypes in turn for predicting data $\hat{F}$ by assignment yields

$$\hat{F} = WF^* = W(W^\top W)^{-1}W^\top F = \Pi_{\mathcal{R}(W)}F = A_1(W)F. \tag{3.28}$$

Finally, optimizing the assignments W in order to obtain the best prediction of the data itself, gives with $A_1(W)^2 = A_1(W)$

$$\operatorname*{arg\,min}_{W\in\mathcal{W}^c} \frac{1}{2}\|A_1(W)F - F\|_F^2 = \operatorname*{arg\,max}_{W\in\mathcal{W}^c} \operatorname{tr}\big(A_1(W)FF^\top\big), \tag{3.29}$$

and the initial assumption of Euclidean data can be dropped by replacing the Euclidean Gram matrix $FF^\top$ by a general inner product matrix $K_{\mathcal{F}}$ corresponding to the embedding of the data into a reproducing kernel Hilbert space.

As a result, the relaxation of problem (3.9) due to Definition 3.2 can be interpreted as finding the best c-dimensional subspace $\mathcal{R}(W)$ spanned by the (soft) indicator vectors of the c clusters (column vectors of W) for self-prediction of the given data.

Another related 'spectral' interpretation results from rewriting the objective in the form

$$\begin{aligned} \operatorname{tr}\big(A_1(W)K_{\mathcal{F}}\big) &= \operatorname{tr}\big(W(W^\top W)^{-1}W^\top K_{\mathcal{F}}\big) = \operatorname{tr}\big((W^\top W)^{-\frac{1}{2}}W^\top K_{\mathcal{F}}W(W^\top W)^{-\frac{1}{2}}\big) \qquad \text{(3.30a)} \\ &= \operatorname{tr}\big(Y(W)^\top K_{\mathcal{F}}Y(W)\big), \qquad Y(W) = W(W^\top W)^{-\frac{1}{2}}. \qquad \text{(3.30b)} \end{aligned}$$

We conclude from Prop. 3.2 that $Y(W)$ varies over the compact Stiefel manifold,

$$Y(W) \in \operatorname{St}(c,n) = \{X \in \mathbb{R}^{n\times c} \colon X^\top X = I_c\}, \tag{3.31}$$

and that the objective (3.30) is the Rayleigh quotient whose maximizer Y spans the subspace of the c dominant eigenvectors of $K_{\mathcal{F}}$ [HM96, Ch. 1]. Note, however, that $Y(W)$ cannot vary freely but is parameterized by $W \in \mathcal{W}^c$.

Difference between $A_0(W)$ and $A_1(W)$. $A_1(W)$ differs from $A_0(W)$ in that the normalizing matrix $C(W)$ of the former self-assignment matrix is replaced by $W^\top W$ in the latter. A consequence due to Prop. 3.2 is that $A_1(W)$ is no longer doubly stochastic and may have negative entries. Hence the probabilistic interpretation (3.22) of the factorization of $A_0(W)$ no longer holds for $A_1(W)$. On the other hand, unlike $A_0(W)$, matrix $A_1(W)$ has fixed rank c and embeds data in a corresponding subspace.

Formulas (3.24) and (3.27) for the formation of latent prototypes (Euclidean case) are the same when using $A_0(W)$ or $A_1(W)$, up to the different normalizing matrices. And how these prototypes are used to represent the data is made explicit by Figure 3.1 and Eq. (3.28), respectively. Both matrices $A_0(W)$ and $A_1(W)$ are equivalent for labelings $W \in \mathcal{W}^c_*$.

3.3.4. *Cluster-Confusion Matrix.* Using (3.18) and (3.21) the entries of the cluster-confusion matrix (3.17) take the form

$$B_{j,l}(W) := \sum_{i\in\mathcal{I}} P(l|i)Q(i|j) = \big(C(W)^{-1}W^\top W\big)_{j,l}, \qquad j,l \in \mathcal{J}. \tag{3.32}$$

This expression may be interpreted as probability that clusters $\mathcal{I}_j$ and $\mathcal{I}_l$ are connected, as opposed to the case of integral assignments (labelings) $W \in \mathcal{W}^c_*$, in which case $B(W) = I_c$ and all clusters are disjoint (hard partition).

4. Self-Assignment Flows

In this section, we generalize the assignment flow (2.36) to the unsupervised scenario discussed in Section 3. Generalizing the likelihood map (2.33) is the major step (Section 4.1). The remaining components of the assignment flow remain unchanged, except for starting the flow at the perturbed barycenter of the assignment manifold in order to break the symmetry through the data, in the absence of labels and any prior information (Section 4.2). Next, we complement in Section 4.3 the interpretations of the relaxations underlying the self-assignment flow (Section 3.3) and show that the latent prototypes determined by the flow maximize class separability. Finally, numerical aspects are discussed in Section 4.4.

4.1. **Generalized Likelihood Map.** In the supervised case, for a given distance matrix $D_{\mathcal{F}}$ (2.31), *local* label assignment is simply achieved by determining separately the smallest component of the vectors $D_{\mathcal{F};i}$, for every vertex $i \in \mathcal{I}$. This corresponds to solving

$$\min_{W \in \mathcal{W}} \operatorname{tr}(D_{\mathcal{F}} W^{\top}) \tag{4.1}$$

and the likelihood map (2.33) lifts the scaled negative *gradient* of this objective function to $\mathcal{S}$. In view of problem (3.9) and the family of self-assignment matrices due to Definition 3.16, a natural approach to generalize this supervised set-up to the unsupervised case is to consider the problem

$$\max_{W} \; E_s(W) \quad \text{subject to} \quad W \in \begin{cases} \mathcal{W}, & \text{if } s = 0 \\ \mathcal{W}^c, & \text{if } s \in (0,1] \end{cases} \tag{4.2a}$$

$$E_s(W) = \operatorname{tr}\big(K_{\mathcal{F}} A_s(W)\big) \tag{4.2b}$$

and to replace $-D_{\mathcal{F}}$ in the likelihood map by the gradient $\partial E_s(W)$. For $s = 0$ and $s = 1$, we have

$$\partial E_0(W) = 2K_{\mathcal{F}} W C(W)^{-1} - \mathbb{1}_n \operatorname{diag}\big(C(W)^{-1} W^{\top} K_{\mathcal{F}} W C(W)^{-1}\big)^{\top}, \tag{4.3a}$$

$$\partial E_1(W) = 2\big(I_n - A_1(W)\big) K_{\mathcal{F}} W (W^{\top} W)^{-1}. \tag{4.3b}$$

In order to substantiate this approach, we interpret these gradients using the concepts from Section 3.3. For illustration, let $K_{\mathcal{F}} = F F^{\top}$ be a Euclidean inner product matrix, with F given by (3.25). Equation (3.24) determining the latent prototypes as averages weighted by the likelihood $Q(i|j)$, Eq. (3.21), reads

$$f_j^* = \sum_{i \in \mathcal{I}} \big(C(W)^{-1} W^{\top}\big)_{j,i} f_i = \big(C(W)^{-1} W^{\top} F\big)_j, \qquad (F^*)^{\top} = F^{\top} W C(W)^{-1}. \tag{4.4}$$

We have

$$\partial E_0(W) = 2 F F^{\top} W C(W)^{-1} - \mathbb{1}_n \operatorname{diag}\big((F^{\top} W C(W)^{-1})^{\top} F^{\top} W C(W)^{-1}\big)^{\top} \tag{4.5a}$$

$$= 2F(F^*)^{\top} - \mathbb{1}_n \operatorname{diag}(F^*(F^*)^{\top})^{\top}, \tag{4.5b}$$

$$\big(\partial E_0(W)\big)_{i,j} = 2\langle f_i, f_j^* \rangle - \|f_j^*\|^2 = -\|f_i - f_j^*\|^2 + \|f_i\|^2, \tag{4.5c}$$

where the prototypes $f_j^* = f_j^*(W)$ depend on W. The last term on r.h.s. of (4.5c) does not depend on j and hence is factored out – cf. Remark 2.1 – when lifting the vector (4.5c) to the assignment manifold as follows. Hence, we ignore this term and generalize the likelihood map (2.33) to

$$L_{0;i}(W_i) = \exp_{W_i}\Big(\frac{1}{\rho} \partial E_0(W)_i\Big) = \exp_{W_i}\Big(-\frac{1}{\rho}\big(\|f_i - f_j^*\|^2\big)_{j \in \mathcal{J}}\Big), \tag{4.6}$$

which amounts to replace the distance vectors $D_{\mathcal{F};i}$, for *given* prototypes in the supervised case, by a *varying* squared distance depending on *latent* prototypes, that emerge when the assignments $W(t)$ follow the assignment flow.

Now let $s = 1$. We return to the 'spectral' interpretation in terms of (3.30) and (3.31). The Riemannian gradient of the Rayleigh quotient $E_1(Y) = \operatorname{tr}(Y^\top K_{\mathcal{F}} Y)$ over the compact Stiefel manifold (3.31) equipped with the standard Euclidean metric reads [AMS09, Sec. 4.8])

$$\operatorname{grad} E_1(Y) = 2(I_n - YY^\top) K_{\mathcal{F}} Y \quad \in \quad T_Y \operatorname{St}(c,n). \tag{4.7}$$

Next we relate the Euclidean gradient (4.3b) to the Riemannian gradient (4.7), taking into account the parametrization $Y(W) \in \operatorname{St}(c,n)$ in (3.30), to obtain

$$\begin{aligned}
\partial E_1(W) &= 2\big(I_n - A_1(W)\big) K_{\mathcal{F}} W (W^\top W)^{-1} && \text{(4.8a)}\\
&= 2\big(I_n - Y(W)Y(W)^\top\big) K_{\mathcal{F}} Y(W) (W^\top W)^{-\frac{1}{2}} && \text{(4.8b)}\\
&= \operatorname{grad} E_1(Y(W)) (W^\top W)^{-\frac{1}{2}}. && \text{(4.8c)}
\end{aligned}$$

Since the second factor in (4.8c) is non-singular, we conclude

$$\partial E_1(W) = 0 \quad \Leftrightarrow \quad \operatorname{grad} E_1(Y(W)) = 0. \tag{4.9}$$

In words, $W \in \mathcal{W}^c$ is a stationary point if and only if $Y(W) \in \operatorname{St}(c,n)$ is a stationary point of the Rayleigh quotient over the compact Stiefel manifold. Consequently the gradient (4.3b) is directly linked to the search direction on the compact Stiefel manifold, in order to determine the invariant subspace corresponding to the c dominant eigenvectors of $K_{\mathcal{F}}$.

As a consequence of these considerations, we define for arbitrary $s \in [0,1]$ the **generalized likelihood map** as

$$L_{s;i}(W_i) = \exp_{W_i}\Big(\frac{1}{\rho} \partial E_s(W)_i\Big), \tag{4.10}$$

with $E_s(W)$ given by (4.2).

4.2. **Self-Assignment Flows.** Besides replacing the likelihood map (2.33) by the generalized likelihood map (4.10), no further changes are required in order to generalize the assignment flow (2.36) to the unsupervised case, except for the initialization which cannot both start at the barycenter and break the symmetry, without any prior information. This will be achieved by taking a small perturbation of the barycenter as initial point.

Accordingly, we define the one-parameter family of **self-assignment flows**

$$\dot{W} = R_W S(W), \qquad W(0) = \exp_{\mathbb{1}_{\mathcal{W}}}(-\varepsilon D_{\mathcal{F},0}), \quad 0 < \varepsilon \ll 1 \tag{4.11a}$$

$$W(t) \in \begin{cases} \mathcal{W}, & \text{if } s = 0, \\ \mathcal{W}^c, & \text{if } s \in (0,1]. \end{cases} \tag{4.11b}$$

The matrix $D_{\mathcal{F},0}$ is computed using the given data $\mathcal{F}_n$ as explained in Section 2.6. The flow $W(t)$ is restricted to the submanifold of full-rank assignments if $s > 0$.

Proposition 3.2 and Eq. (3.13) yield the following.

Corollary 4.1. *Let $W(t)$ solve (4.11). Then, for any $t \geq 0$,*

(i) the self-affinity matrix $A_0\big(W(t)\big)$ is doubly stochastic and completely positive, if $s = 0$;

(ii) the self-influence matrix $A_1\big(W(t)\big)$ is iso-spectral, i.e. its eigenvalues satisfy $\lambda_1 = \cdots = \lambda_c = 1$ and $\lambda_{n-c} = \cdots = \lambda_n = 0$, if $s = 1$.

(iii) $A_0\big(W(T)\big) = A_1\big(W(T)\big)$ if $W(T) \in \mathcal{W}^c_*$.

Property (iii) relates to the fact that $W(t)$ solving (4.11) approaches a labeling $W(T) \in \mathcal{W}^c_*$ for sufficiently large T after a trivial rounding step. We point out, however, that solving (4.11) generally yields different paths $W(t)$, $t \in [0,T]$ depending on $s \in [0,1]$ and corresponding to the different relaxations, as discussed in Section 3.3. Once a labeling $W(T) \in \mathcal{W}_*^c$ has been computed, using any $s \in [0,1]$, the solution is a local optimum of the partitioning problem (3.9). This is what Corollary 4.1(iii) says.

Remark 4.1. All these considerations remain valid with the effective number $\hat{c}$ of clusters in place of c, if $\hat{c}$ should be smaller than c; see Definition 4.1 below.

4.3. **Self-Assignment Performs Self-Supervision.**

We interpret the assignment flow from another angle that complements the interpretations discussed in Section 3.3.

In Section 3.3.2, we showed that running the assignment flow entails learning of latent prototypes that can be explicitly recovered if weighted means in the data space are well-defined and computationally feasible. Let us temporarily adopt the Euclidean situation (3.24). With these recovered prototypes at hand, we get back to Section 2.2 and ask how our approach relates to the *supervised* situation where the quality of the clustering can be assessed by objectives like (2.6). Assuming a labeling $W = W(T) \in \mathcal{W}_*^c$ has been determined, let the recovered prototypes f_j^*, $j \in \mathcal{J}$ play the role of the empirical means m_j, $j \in \mathcal{J}$. We compute in terms of the data matrix F (3.25) the quantities (2.3)

$$P_j = \frac{1}{n}\langle W^j, \mathbb{1}_n\rangle = \frac{1}{n}|\mathcal{I}_j|, \qquad j \in \mathcal{J} \qquad \text{(prior probabilities)} \tag{4.12a}$$

$$f_j^* = F^\top\big(WC(W)^{-1}\big)^j, \qquad j \in \mathcal{J} \qquad \text{(class-conditional mean vectors)} \tag{4.12b}$$

$$f^* = \frac{1}{n}F^\top \mathbb{1}_n, \qquad \text{(mean vector)} \tag{4.12c}$$

and in turn the scatter matrices (2.4)

$$S_t = \frac{1}{n}\sum_{i\in\mathcal{I}}(f_i - f^*)(f_i - f^*)^\top = \frac{1}{n}F^\top\big(I - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top\big)F, \tag{4.13a}$$

$$S_w(W) = \frac{1}{n}\sum_{j\in\mathcal{J}}\sum_{i\in\mathcal{I}_j}(f_i - f_j^*)(f_i - f_j^*)^\top = \frac{1}{n}F^\top\big(I - A_0(W)\big)F, \tag{4.13b}$$

$$S_b(W) = \sum_{j\in\mathcal{J}} P_j(f_j^* - f^*)(f_j^* - f^*)^\top = \frac{1}{n}F^\top\big(A_0(W) - \frac{1}{n}\mathbb{1}_n\mathbb{1}_n^\top\big)F. \tag{4.13c}$$

Regarding the dependency on W, we observe that the within-class scatter matrix $S_w(W)$ involves the term $F^\top A_0(W)F$ and the between-class scatter $S_b(W)$ the term $-F^\top A_0(W)F$. Hence, by minimizing the objective (3.9), we *simultaneously* minimize $\operatorname{tr}(S_w)$ and maximize $\operatorname{tr}(S_b)$:

$$\operatorname*{arg\,min}_W \operatorname{tr}\big(S_w(W)\big) \quad \Leftrightarrow \quad \operatorname*{arg\,max}_W \operatorname{tr}\big(S_b(W)\big) \quad \Leftrightarrow \quad \operatorname*{arg\,max}_W \operatorname{tr}\big(A_0(W)FF^\top\big). \tag{4.14}$$

We conclude that the latent prototypes determined by the self-assignment flow turns a completely unsupervised scenario into a supervised one, in agreement with established measures for class separability like (2.6). This interpretation also remains valid when the relaxation with $s = 1$ and objective (3.29) is used to compute a labeling W, due to Corollary 4.1(iii).

Moreover, since the approach only depends on the inner product matrix $FF^\top$, it generalizes to data embeddings into a reproducing kernel Hilbert space and a corresponding data affinity matrix $K_{\mathcal{F}}$ with entries (3.2).

Remark 4.1 applies to the above considerations. Just replace below $(c, \mathcal{J})$ by $(\hat{c}, \hat{\mathcal{J}})$.

4.4. **Geometric Numerical Integration.** We distinguish the two cases (4.11b).

Case $s = 0$. We directly apply the methods studied by [ZSPS19]. To make this paper self-contained, we merely state the simplest scheme, the geometric Euler method. This explicit scheme with fixed step-size $h > 0$ reads

$$W_i^{(k+1)} = \operatorname{Exp}_{W_i^{(k)}}\big(hR_{W_i^{(k)}}S(W^{(k)})\big), \quad i \in \mathcal{I}. \tag{4.15}$$

It ensures that the self-assignment flow (4.11a) evolves properly on the assignment manifold $\mathcal{W}$. The iteration (4.15) stops when the average entropy of the assignments $W^{(K)}$ drops at some iteration $k = K$ below the predefined threshold 10^{-3}, which indicates (almost) unique label assignments and hence stationarity of the flow evolution. Then numerical integration is terminated and a labeling $W \in \mathcal{W}_*^{\hat{c}}$, $\hat{c} \leq c$, is determined using $W^{(K)}$ in a trivial postprocessing step by selecting the most likely label for each row $W_i^{(K)}$, $i \in \mathcal{I}$ and removing the $c - \hat{c}$ zero-columns (corresponding to empty clusters) from the resulting labeling $W \in \mathcal{W}_*^{\hat{c}}$.

Definition 4.1 (**Effective Number $\hat{c}$ of Clusters (Labels)**)**.** We call the just described number

$$\hat{c} \leq c \tag{4.16}$$

the *effective number of clusters or labels*, respectively. It is determined by the homogeneity of the data $\mathcal{F}_n$ and by the scale

$$|\mathcal{N}_i|, \quad i \in \mathcal{I} \qquad\qquad (\textbf{scale}) \tag{4.17}$$

at which regularization is performed by the assignment flow through the similarity map (2.35). We denote the corresponding index set of labels by

$$\hat{\mathcal{J}} \subset \mathcal{J}, \qquad |\hat{\mathcal{J}}| = \hat{c}. \tag{4.18}$$

Case $s = 1$. Integration of the self-assignment flow (4.11a) restricted to the open submanifold $\mathcal{W}^c$ of full-rank assignments (3.14) is more involved. Corresponding geodesics only locally exist on $\mathcal{W}$, i.e. full-rank assignment matrices cannot be guaranteed during the numerical integration process (4.15). Clearly, if the data affinity matrix $K_{\mathcal{F}}$ has high rank (induced by heterogeneous data) and if the scale (4.17) for regularization is not chosen too large, a full-rank labeling $W \in \mathcal{W}^c$ may be returned by the self-assignment flow, that is well-defined in view of the relation (4.9).

In order to handle other cases while still using the numerical scheme (4.15) or more sophisticated ones [ZSPS19], we simply replace the inverse normalizing matrix by its pseudo-inverse,

$$(W^\top W)^{-1} \quad \longleftarrow \quad (W^\top W)^\dagger. \tag{4.19}$$

Whenever this regularization of the normalizing matrix becomes 'active', we extract the effective number $\hat{c}$ in a postprocessing step, as described above in the case $s = 0$.

5. Related Work and Discussion

The literature on clustering is vast. We therefore restrict the discussion to few major methodological directions in the literature: Graph cuts and spectral relaxation (Section 5.1), discrete regularized optimal transport (Section 5.2) and combinatorial optimization for graph partitioning (Section 5.3).

5.1. **Graph Cuts and Spectral Relaxation.** Summing up the weights (affinities) of edges that are cut provides a natural quality measure for graph partitioning. To avoid unbalanced partitions, such measures are normalized in various ways, and spectral relaxations of the resulting combinatorial optimization problem renders the computation of good suboptimal solutions feasible. We refer to [vL07] for a survey.

We focus on two basic balanced cut-criteria that can be expressed by the graph Laplacian

$$L_{\mathcal{F}} = D_{K,\mathcal{F}} - K_{\mathcal{F}}, \qquad D_{K,\mathcal{F}} = \operatorname{Diag}(K_{\mathcal{F}}\mathbb{1}_n) \tag{5.1}$$

and indicator vectors. The *ratio-cut criterion* reads

$$\min_{U\in\mathbb{R}^{n\times c}} \operatorname{tr}(U^{\top}L_{\mathcal{F}}U) \quad \text{subject to} \quad U \geq 0, \quad U^{\top}U = I_c, \tag{5.2}$$

whereas the *normalized-cut (Ncut) criterion* [SM00] additionally uses the degree matrix $D_{K,\mathcal{F}}$ for normalization,

$$\min_{U\in\mathbb{R}^{n\times c}} \operatorname{tr}(U^{\top}L_{\mathcal{F}}U) \quad \text{subject to} \quad U \geq 0, \quad U^{\top}D_{K,\mathcal{F}}U = I_c. \tag{5.3}$$

Due to the conjunction of nonnegativity and orthogonality constraints, both problems (5.2) and (5.3) are difficult to optimize globally. *Spectral relaxation* means to drop the element-wise nonnegativity constraint. Then the relaxed problems (5.2) and (5.3) amount to solving an eigenvalue problem and a generalized eigenvalue problem, respectively. The price to pay in either case is that the physical interpretation of U as indicator variables is lost and must be recovered by an additional post-processing step, which is usually done by applying the classical k-means algorithm.

A direct relation to the proposed self-assignment flow is apparant in the case $s = 1$. Substituting $Y = D_{K,\mathcal{F}}^{1/2}U$ in the spectral relaxation of (5.3) results in the problem

$$\max_{Y\in\mathbb{R}^{n\times c}} \operatorname{tr}\left(Y^{\top}\tilde{K}_{\mathcal{F}}Y\right) \quad \text{subject to} \quad Y^{\top}Y = I_c, \tag{5.4}$$

that is, the Rayleigh quotient of the *normalized* affinity matrix $\tilde{K}_{\mathcal{F}} = D_{K,\mathcal{F}}^{-1/2}K_{\mathcal{F}}D_{K,\mathcal{F}}^{-1/2}$ has to be maximized over the compact Stiefel manifold (3.31). As already discussed for $s = 1$ in connection with (3.30), assignments W following the self-assignment flow parametrize points $Y(W) \in \operatorname{St}(c,n)$ on the compact Stiefel manifold that maximize the Rayleigh quotient: Eq. (4.8c) shows that the driving force of the self-assignment flow (generalized likelihood map) is directly linked to the gradient ascent of the Rayleigh quotient over the compact Stiefel manifold. Finally, when the numerical integration of the self-assignment flow terminates, then the resulting labeling $W \in \mathcal{W}_*^c$ together with (3.13) ensures $Y(W) \geq 0$. Hence, after re-substitution, $U(W) = D_{K,\mathcal{F}}^{-1/2}Y(W)$ is directly feasible for the original problem (5.3) and hence no 'projection' by k-means is required as post-processing.

The common way to take into account *spatial regularization* in spectral clustering is to augment given features by spatial coordinates. However, this strategy suffers from a conceptual shortcoming, since augmentation makes the *same* feature vector differ when it is observed at two different spatial locations. In contrast, the self-assignment flow performs unbiased spatial regularization by smooth geometric averaging and recognizes closeness of features no matter *where* they are observed.

5.2. **Discrete Regularized Optimal Transport.** The theory of optimal transport [Vil09, San15] has become a major modeling framework for data analysis. Here we focus on discrete optimal transport and computational aspects [BCPD99, Pey18].

We consider the case $s = 0$ and the self-affinity matrix $A_0(W)$. Since $A_0(W)$ is doubly stochastic (Prop. 3.2), maximizing the objective $E_0(W)$ (4.2b) may be interpreted as a discrete optimal transport problem with cost matrix $K_{\mathcal{F}}$ and uniform marginal measures (3.19). These marginals correspond to the data $\mathcal{F}_n$ and a copy of the data, respectively, resulting in data self-assignment as discussed in Section 3.3.1.

For further interpretation, we consider the Euclidean case $K_{\mathcal{F}} = FF^\top$. Inserting the explicit form (3.12) of $A_0(W)$ into the objective $E_0(W)$ and using (4.4), we obtain

$$E_0(W) = \operatorname{tr}(K_{\mathcal{F}} W C(W)^{-1} W^\top) = \operatorname{tr}(W F^* F^\top). \tag{5.5}$$

Maximizing this objective function reveals what this problem relaxation actually means: A linear assignment problem in terms of the assignment matrix W with *varying* inner product matrix $F^*(W)F^\top$ as costs. Moreover, since $W \in \mathcal{W}$, we have a fixed marginal $W\mathbb{1}_c = \mathbb{1}_n$ and a the second marginal $W^\top \mathbb{1}_n = \operatorname{diag}\big(C(W)\big)$ which is *free*. Alltogether, a quite difficult problem is solved in terms of W: latent prototypes F^* are formed by *transporting* the uniform prior measure to the support of the respective clusters, so as to maximize the correlation $E_0(W)$ of the assignments W and the inner product matrix $F^*F^\top$.

We point out a key property of the assignment flow that makes this approach work: It is the spatial regularization performed by the similarity map (2.35) that drives the entire process, in addition to the underlying geometry that makes $W(t)$ converge towards hard assignments (labelings). In fact, without spatial regularization, the self-affinity matrix $A_0(W) = I_n$ would maximize $E_0(W)$ assuming the similarity $k_{\mathcal{F}}(f_i, f_k)$ is maximal if $f_i = f_k$, which means that every given data point f_i forms its own cluster. This trivial solution is ruled out, by construction, through the factorization with rank upper bounded by c and through geometric spatial averaging of the assignments. The corresponding *scale* in terms of the sizes of the neighborhoods (2.18) determines how coarse or fine the spatial arrangement of the resulting clusters will be.

We informally summarize this discussion: Data self-assignment is defined by uniform marginal measures and a coupling measure parametrized by the assignment flow. Structure in the data is induced by imposing a low-rank constraint (factorization) on the coupling measure (transport plan) and through spatial regularization of the flow of assignments.

5.3. **Combinatorial Optimization.** Zass and Shashua [ZS05] studied the formulation of the clustering problem

$$\max_{W \in \mathbb{R}^{n \times c}} \operatorname{tr}(K_{\mathcal{F}} W W^\top) \qquad \text{subject to} \tag{5.6a}$$

$$\text{(a) } W \geq 0, \qquad \text{(b) } \operatorname{rank}(W) = c, \qquad \text{(c) } W^\top W = I_c, \qquad \text{(d) } W W^\top \mathbb{1}_n = \mathbb{1}_n \tag{5.6b}$$

in terms of the completely positive factorization $WW^\top$ and the constraints (a)–(d). We notice that the orthogonality constraint (c) with respect to the columns of W implies (b), and that (a) together with (d) says that $WW^\top$ is doubly stochastic. The authors show that (a)–(d) imply that $W \in \mathcal{W}_*^c$ is a labeling. This problem formulation differs from more classical conditions ensuring $W \in \mathcal{W}_*^c$ [RW95, Lemma 2.1],

$$W \geq 0, \qquad W\mathbb{1}_c = \mathbb{1}_n, \qquad W^\top \mathbb{1}_n = (n_1, \dots, n_c)^\top, \qquad \operatorname{tr}(W^\top W) = n, \tag{5.7}$$

in that the cluster sizes (third constraint) do not have to be specified beforehand.

Regarding relaxation, the authors of [ZS05] argue that the orthogonality constraint (c) is the weakest one. They propose a two-step procedure after dropping the constraints (b) and (c): approximation of the data similarity matrix $K_{\mathcal{F}}$ by a doubly stochastic matrix using the Sinkhorn iteration, followed by a gradient ascent iteration with stepsize control so as to respect the remaining constraints. The same set-up was proposed by [YC16] except for determining a locally optimal solution by a single iterative process using DC-programming. Likewise, [KYP15] explored symmetric nonnegative factorizations but ignored the constraint enforcing that $WW^\top$ is doubly-stochastic, which is crucial for cluster normalization.

Our approach uses the factorization $A_s(W)$ given by (3.16) instead of $WW^{\top}$. While the orthogonality constraint (c) is dropped as well, the constraints (a) and (d) are 'built in' by construction, and constraint (b) may additionally hold (cf. Definition 4.16 and the corresponding discussion). Furthermore, optimization is achieved by a single *smooth and continuous* process, the self-assignment flow (4.11), which enables to apply numerous discrete numerical schemes [ZSPS19], all of which respect the constraints. Finally, geometric regularization within local neighborhoods of each vertex of the underlying graph through the similarity map (2.35) enforces the formation of 'natural' clusters, whenever assigning the same label to close vertices is more likely to be correct.

6. Experiments

In this section, we demonstrate and evaluate the performance of the proposed one-parameter family (4.11) of *self-assignment flows (SAF)* for unsupervised data labeling, using various datasets and feature spaces (Figure 6.1).

After describing specific details of the implementation (Section 6.1), we report the study of the two model parameters in Section 6.2, and the influence of affinity matrix sketching for data reduction in a preprocessing step, to make learning from large data sets computationally feasible. In Section 6.3, we compare our approach to various methods: basic clustering, normalized spectral cuts with spatial regularization, and partitioning using a variational decomposition of the piecewise constant Mumford-Shah model. We focus on an attractive application of our approach in Section 6.4: Learning patch dictionaries using the SAF based on a locally invariant distance function. Finally, as a sanity check, we report the application of the SAF to problem data on a graph from a domain that is unrelated to image analysis, to substantiate our claim that our approach applies to any data given on any graph, in principle.

6.1. **Implementation Details.** Throughout this paper, the SAF (4.11) was numerically integrated using the geometric explicit Euler scheme (4.15) with step-size $h = 0.1$, as described in Section 4.4. For parameter values $s \in (0,1]$, we applied (4.19) to avoid numerical problems when the effective number of clusters $\hat{c} < c$ (Def. 4.1) actually was smaller than c. The SAF with $s = 0$ does not encounter any such problems, due to the different normalization involved in (3.12). We adopted from [ÅPSS17] the numerical renormalization step for the assignments with $\varepsilon = 10^{-10}$, to avoid numerical issues for assignments very close to the boundary of the assignment manifold. Numerical integration was terminated when the average entropy of the assignments dropped below the threshold of 10^{-3}, which indicates that the current iterate is very close to an almost unique assignment (labeling) $W^{(k)} \in \mathcal{W}_*^{\hat{c}}$.

Unless specified otherwise, the default value $\rho = 0.1$ (distance normalization in (2.33)) and uniform weights $w_{i,k} = 1/|\mathcal{N}_i|$ (2.34) for assignment regularization were used in all experiments, with neighborhoods $\mathcal{N}_i$ of equal size

$$|\mathcal{N}| := |\mathcal{N}_i|, \quad \forall i \in \hat{\mathcal{I}}, \tag{6.1}$$

for interior pixels $\hat{\mathcal{I}} \subset \mathcal{I}$.

Data $\mathcal{F}_n$ were embedded using the standard Gaussian kernel (3.3) with parameter $\sigma = \sqrt{0.1}$, in order to compute the affinity matrix $K_{\mathcal{F}}$ (3.2). For larger datasets, a sketch of $K_{\mathcal{F}}$ was used as described in Section 2.3, with parameters $q = 1$ and $\ell = 100$ random samples drawn without replacement; see Section 6.2 for a validation. Finally, the initial value $W(0)$ of (4.11a) was chosen as small perturbation of the barycenter (4.11a) with $\varepsilon = 10^{-2}$ and initial distance matrix $D_{\mathcal{F},0}$, computed with the inexpensive greedy k-center clustering algorithm, as explained in Section 2.6.

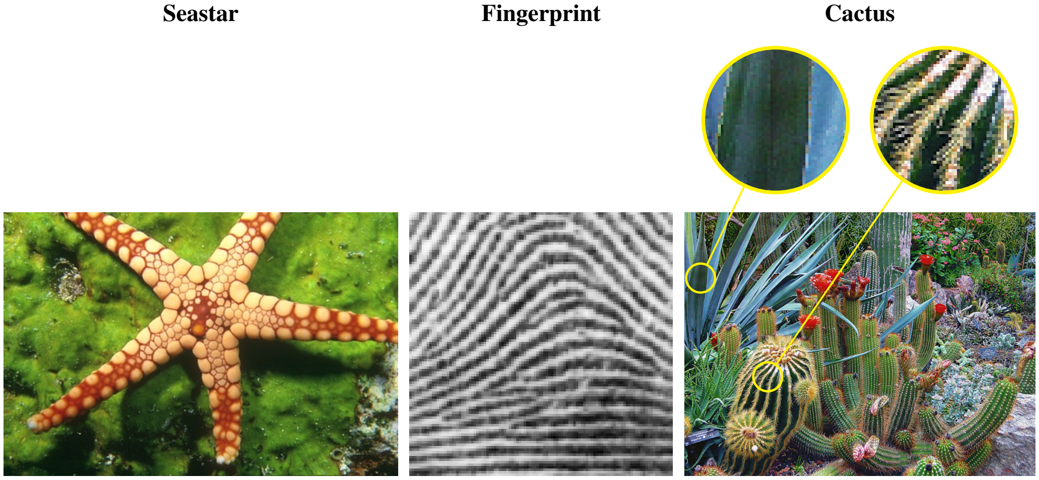


FIGURE 6.1. Input image data used in the numerical experiments (Fig. 6.2, 6.5, 6.6, 6.8, 6.9). Close-up views enable to compare the influence of model parameters on local image structure in comparison to alternative approaches from related work. Both the Euclidean RGB-space and locally invariant patch spaces were used as feature spaces. Regarding the latter, additional real image data are processed in Figures 6.10 and 6.11. The results of graph network data are depicted by Figure 6.12 in order to highlight that our approach more generally applies to data on graphs, beyond image feature data.

6.2. **Influence of Model Parameters.** The self-assignment flow (SAF) has three model parameters: The parameter s of the self-assignment matrix $A_s(W)$ (3.16a), the neighborhood size $|\mathcal{N}|$ controlling the *scale* of regularization, and the upper bound c on the effective number $\hat{c}$ of labels (4.16).

6.2.1. *Influence of s, $|\mathcal{N}|$ and c.* Figure 6.2 shows both labelings and recovered prototypes below each panel, depending on s and $|\mathcal{N}|$. We set $c = 16$ which is sufficiently large, since $\hat{c} < c$ quickly happens when lowering s even at the smallest scale of 3×3 pixels. $\hat{c}$ further drops down with larger scale. Regarding the parameter s, we observe:

- **Small s:** Spatial regularization is more aggressively enforced, leading to compact codes in terms of smaller numbers $\hat{c}$ of prototypes.
- **Large s:** Distances in the feature space have more impact. Local image structure is better preserved at the cost of a larger number $\hat{c}$ of prototypes.

The second observation underlines the relation of the self-assignment flow, for $s = 1$, to spatially regularized normalized cuts as worked out in Section 5.1.

Figure 6.2 illustrates that depending on the application, the properties of the SAF can be continuously controlled by setting the parameter s, thanks to the geodesic interpolation (3.16).

6.2.2. *Evolution of Cluster Sizes, Entropy, and Rank Lower Bound.* Figure 6.3 illustrates the *evolution* of the SAF in terms of the following measurements.

- **Cluster sizes:** For smaller values of s, more iterations are required for cluster formation. This conforms with the observation in Section 6.2.1 that the SAF then promotes spatial regularization. Conversely, larger values of s yield more balanced (uniform) cluster sizes. This is

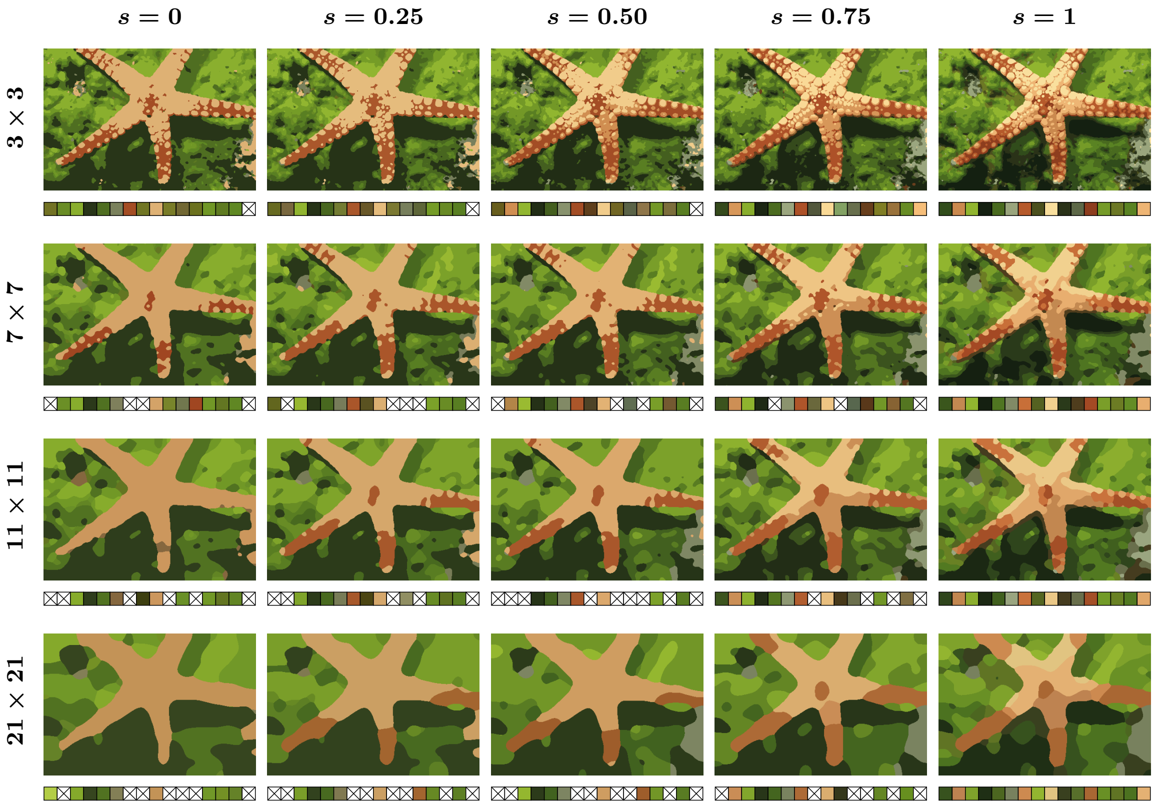


FIGURE 6.2. Influence of the model parameters $s \in [0, 1]$ parametrizing the SAF in terms of the self-assignment matrix (3.16), the neighborhood size $|\mathcal{N}|$ controlling the scale of spatial regularization, and the effective number $\hat{c} \leq c = 16$ of labels. Recovered prototypes are displayed below each labeling and aligned to each other (using linear assignment of the clusters) to ease visual comparison. Prototypes that 'died out' are marked by a cross. We observe that due the geodesic interpolation (3.16), the influence of spatial regularization (small s: compact image codes) relative to the influence of distances in the feature space (large s: preserving local image structure) can be continuously controlled.

consistent with the observation made in Section 6.2.1 that, in this case, the SAF more carefully explores the feature space and preserves local image structure.

Average entropy: The panels illustrate that the initial assignment is an ε-perturbation of the barycenter on the assignment manifold, and that the termination criterion was reached in all experiments. In agreement with the preceding point, the SAF converges faster for larger values of s.

Rank lower bound: The third row of Figure 6.3 displays the lower bound $\operatorname{tr}\big(B(W^{(k)})\big)$ of $\operatorname{rank}(W^{(k)})$ due to Proposition 3.3(d). After termination of the SAF, this lower bound becomes sharp at $W \in \mathcal{W}_*^{\hat{c}}$ and attains the number $\hat{c}$ of effective prototypes.

6.2.3. *Influence of Affinity Matrix Sketching.* We evaluate the influence of sketching the data affinity matrix $K_{\mathcal{F}}$ in a preprocessing step, as described in Section 2.3, using the parameter value $q = 1$ and varying sample sizes ℓ.

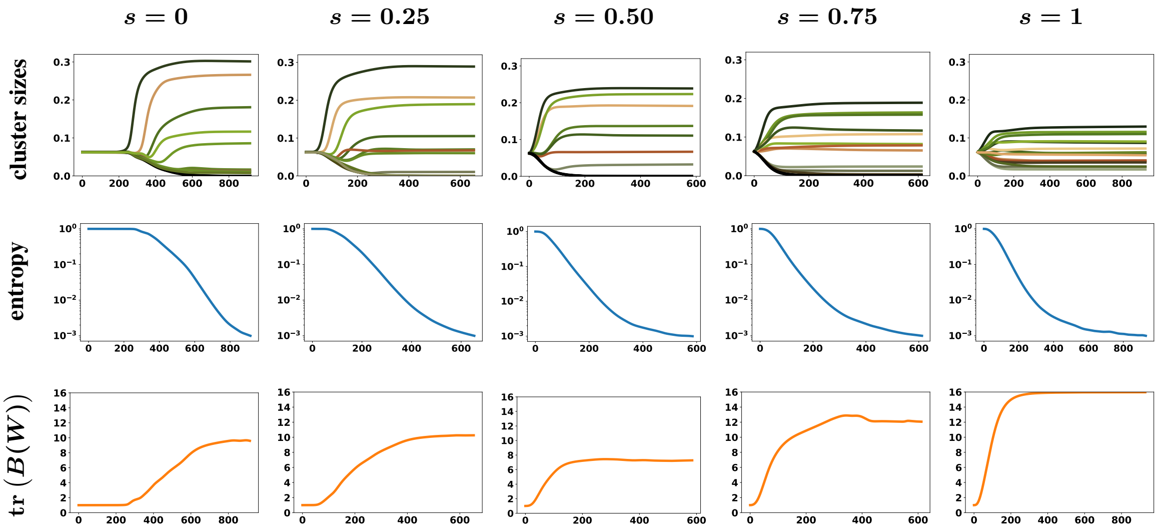


FIGURE 6.3. Evolution of relative cluster sizes, average entropy and lower bound of $\mathrm{rank}(W^{(k)})$ as a function of the SAF, depending on the iterations k for the experiment with $|\mathcal{N}| = 11 \times 11$ depicted by Fig. 6.2. TOP: Smaller values of s promote spatial regularization. Hence more iterations are required to form clusters. Larger values of s yield more uniform cluster sizes which reflects the stronger influence of feature similarity and the preservation of local image structure. CENTER: The average entropy illustrates the random initialization ε-close to the barycenter and that the termination criterion is reached in all experiments. The entropy decays faster for larger values of s. BOTTOM: The lower rank bound due to Proposition 3.3(d) becomes sharp when the SAF terminates at some labeling $W \in \mathcal{W}_{*}^{\hat{c}}$ and attains the number $\hat{c}$ of effective labels.

To this end, we focused on the experiment with $s = 0$, $|\mathcal{N}| = 3 \times 3$ depicted by Figure 6.2 and compared the labelings obtained with and without sketching $K_{\mathcal{F}}$. To handle the latter case where $K_{\mathcal{F}}$ requires ≈ 177 GB of memory, we computed on the fly the entries for every matrix-vector multiplication on GPUs using the software library KeOps[1], rather than holding the matrix in memory.

Figure 6.4 displays the relative error of different label assignments after sketching, depending on the sample size ℓ, where 100% corresponds to all $n = 321 \times 481$ columns of $K_{\mathcal{F}}$. For each value ℓ, 100 runs were made using different random seeds. Figure 6.4 displays the *average* error along with the standard deviation. The corresponding curves show that $\ell = 100$ samples, i.e. merely 0.065% of all data points, suffice to eliminate the effect of data reduction by sketching the input affinity matrix.

6.3. **Comparison to Other Methods.** We compared the SAF to the following methods:

Nearest neighbor clustering: k**-means** and k**-center** clustering (no spatial regularization), to show the influence of spatial regularization performed by the SAF on both labeling and prototype formation;

[1] B. Charlier, J. Feydy, and J.-A. Glauns, *KeOps Kernel Operations on the GPU*, 2018, https://www.kernel-operations.io/keops/index.html

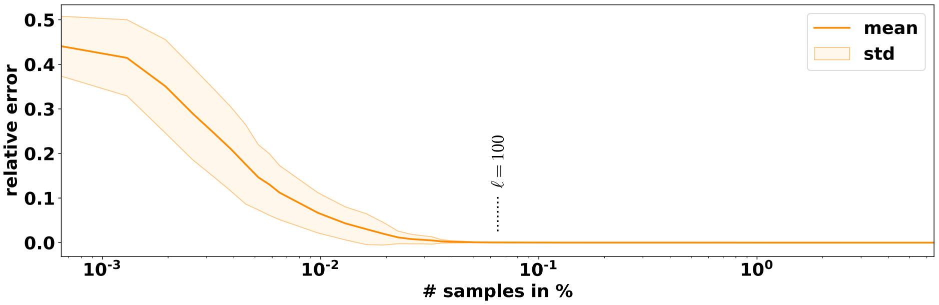


FIGURE 6.4. This plot shows the *average relative labeling error* together with the standard deviation, that result from data reduction by sketching the data affinity matrix $K_{\mathcal{F}}$ in a preprocessing step. for SAF is approximated by the matrix sketching method (see Sec. 2.3) in dependency of the number of sampled pixels l represented in $\%$. The curves show that merely 0.065% of all data points (corresponding to $\ell = 100$ randomly sampled columns of $K_{\mathcal{F}}$) suffice to eliminate the effect of data reduction.

AF: *supervised* assignment flow [ÅPSS17] with spatial regularization, using *fixed* prototypes computed beforehand using nearest neighbor clustering, to highlight that the SAF *simultaneously* performs unsupervised label *learning* and label *assignment*;

Spectral clustering: We computed partitions using normalized spectral cuts [SM00] after augmenting feature vectors by spatial coordinates x_i, $i \in \mathcal{I}$ for spatial regularization. The resulting data affinity matrix was given by

$$K_{\mathcal{F}i,k} = \exp\big(-\big(\tfrac{1}{\sigma^2}\|f_i - f_k\|_2^2 + \alpha\|x_i - x_k\|_2^2\big)\big), \quad i,k \in \mathcal{I}, \tag{6.2}$$

with parameter $\alpha > 0$ controlling the influence of spatial regularization.

Fast partitioning: A variational decomposition of the piecewise-constant Mumford-Shah approach to image partitioning proposed by [SW14], using the publicly available implementation “Pottslab” from the authors. The method operates directly on the values in the feature space instead of using a reformulation with labels. Therefore, the number of clusters can be large. For this reason, we applied an additional k-means clustering step to the (oversegmented) results in order to have a direct comparison in terms of labels and prototypes.

Two variants of the SAF were evaluated for comparison: (i) using *uniform* weights for spatial regularization; (ii) using *nonuniform* weights determined in ”non-local means fashion” by

$$w_{i,k} = \frac{\tilde{w}_{i,k}}{\langle \tilde{w}_i, \mathbb{1}_n\rangle} \quad \text{with} \quad \tilde{w}_{i,k} = \begin{cases} \exp\big(-\frac{1}{\rho}\|P_i - P_k\|_F^2\big), & \text{if } k \in \mathcal{N}_i, \\ 0, & \text{else}, \end{cases} \tag{6.3}$$

where P_i denotes the patch centered at pixel i. Throughout, the patch size as well as the neighborhood size $|\mathcal{N}|$ for geometric averaging was chosen to be 5×5 pixels.

The user parameters of all other methods were manually tuned so as to obtain best comparable results.

6.3.1. *Nearest Neighbor Clustering, Supervised Assignment Flow.* Figure 6.5 displays the results. The close-up view of the results of nearest neighbor clustering shows noisy label assignments even in homogeneous regions, due to the absence of spatial regularization. By contrast, the AF returns spatially coherent labelings. However, since the labels (prototypes) are fixed beforehand, their assignments yield partitions that may locally look unnatural (see close-up views). Note that the prototypes

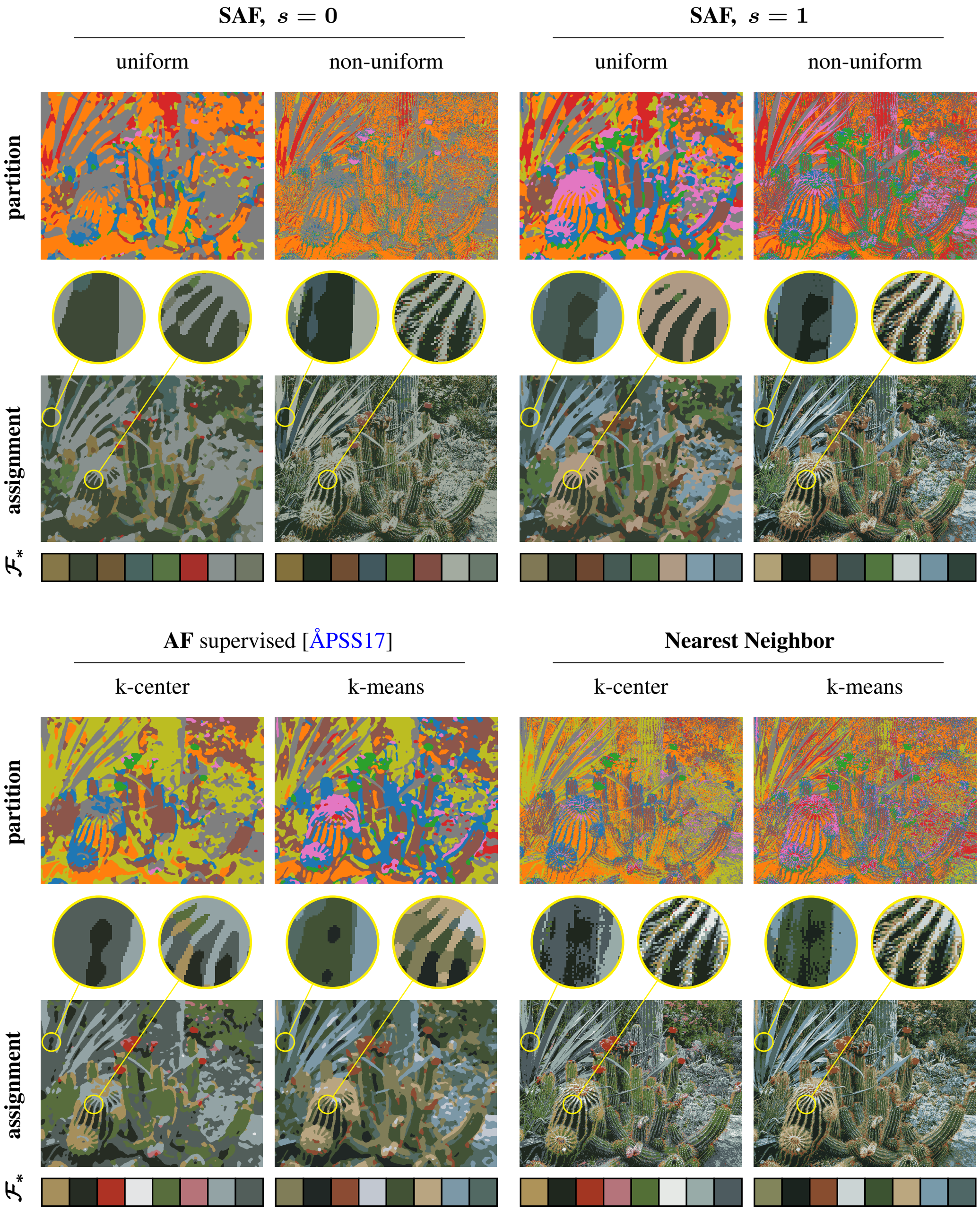


FIGURE 6.5. Comparison of the SAF to nearest neighbor clustering and supervised assignment flow (AF). Inspecting the results and the close-up views shows: Nearest neighbor clustering yields noisy label assignments due to the absence of spatial regularization. The AF returns spatially coherent partitions that may locally look unnatural (see close-up views), since the prototypes are fixed and do not adapt to the spatial components of the resulting partition. The unsupervised SAF learns labels adaptively during label assignment. The resulting partitions have a natural spatial structure with increased details if $s = 1$. The latter effect is considerably enhanced, independent of s, when nonuniform weights are used.

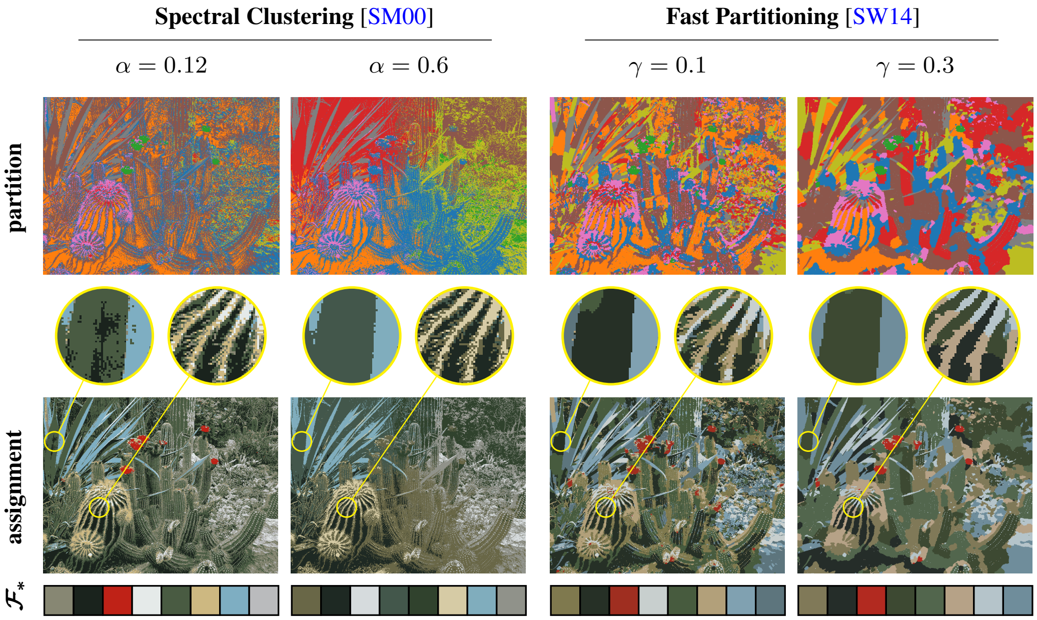


FIGURE 6.6. Comparison of the SAF to spectral clustering using feature vectors augmented by spatial coordinates and normalized cuts, and to fast partitioning that approximates the piecewise constant Mumford-Shah model. Spatial regularization as performed by spectral clustering is clearly suboptimal, since weak regularization returns noisy partitions where strong regularization yields biased clusters (e.g. red cluster). See the last paragraph of Section 5.1 for an explanation. Fast partitioning yields good labelings but does not consistently enforce the scale of spatial regularization through the choice of γ – see, e.g. the small red clusters in the panel on the right-hand side. This reflects that fast partitioning directly operates on the feature space rather then separating data representation from inference, as does the SAF.

displayed for the AF were recomputed after convergence from the resulting partition and, therefore, differ from the nearest neighbor prototypes that were used as input labels for computing the AF.

In comparison with these methods, the SAF yields more natural partitions due to forming the labels *during* label assignment and preserves fine structure for $s = 1$, in agreement with the experiments discussed in Section 6.2. This latter effect is considerably enhanced when nonuniform weights are used, independently of s, without compromising the quality of the spatial structure of the resulting partitions.

6.3.2. *Spatial Feature Augmentation and Normalized Spectral Cuts.* Figure 6.6 displays the corresponding results for spectral clustering and fast partitioning, respectively, using two parameter values enforcing weak and strong spatial regularization in either case.

We observe that spectral clustering is highly sensitive to the value of α. Small values yield noisy partitions, whereas larger values yield biased partitions (e.g. red cluster). We attribute this strange behavior to the conceptual deficiency of spatial regularization performed by feature augmentation, as discussed in the last paragraph of Section 5.1.

Fast partitioning returned the closest labelings to those computed by the SAF. The scale of spatial regularization is not consistently enforced everywhere, however, as e.g. the small red dots on the cactus arms reveal. We attribute this to the above-mentioned fact that fast partitioning directly operates on the feature space, rather than separating data representation from inference using labels and label assignments. In addition, the variational decomposition may be susceptible to getting stuck in suboptimal minima.

6.4. **Unsupervised Learning and Assignment of Locally Invariant Patch Dictionaries.** In this section, we base the self-assignment flow (SAF) on more advanced features, viz. feature *patches*, and a corresponding locally invariant distance function.

6.4.1. *Locally Invariant Patch Distances.* Let

$$\mathcal{N}_{\mathcal{P},i}, \quad i \in \hat{\mathcal{I}}, \qquad\qquad n_{\mathcal{P}} := |\mathcal{N}_{\mathcal{P},i}|, \quad \forall i \tag{6.4}$$

denote quadratic sections centered at pixel (vertex) i of the underlying image grid graph, with uniform size $n_{\mathcal{P}} = 2k+1$ for some $k \in \mathbb{N}$, for every i. We only consider region centers at interior grid points $i \in \hat{\mathcal{I}} \subset \mathcal{I}$ such that no section $\mathcal{N}_{\mathcal{P},i}$ extends beyond the boundary of the graph, which implies

$$\mathcal{N}_{\mathcal{P},i} \subset \mathcal{I}, \quad \forall i \in \hat{\mathcal{I}}. \tag{6.5}$$

We define a *patch centered at pixel* i as the ordered tuple of data points

$$P_i = \big(f_{k_1}, \dots, f_i, \dots, f_{k_{n_{\mathcal{P}}}}\big), \qquad k_1, \dots, k_{n_{\mathcal{P}}} \in \mathcal{N}_{\mathcal{P},i}, \quad i \in \hat{\mathcal{I}}, \tag{6.6}$$

where the particular chosen order does not matter, but should be fixed for all patches. The individual patch features are denoted by

$$P_{i;m} = f_m, \quad m \in \mathcal{N}_{\mathcal{P},i} \tag{6.7}$$

and the collection of all patches induced by the data $\mathcal{F}_n$ is denoted by

$$\mathcal{P}(\mathcal{F}_n) = \big\{P_i \in \mathcal{F}_n^{n_{\mathcal{P}}} \colon\, i \in \hat{\mathcal{I}}\big\}. \tag{6.8}$$

In order to define invariant distance functions, we consider the dihedral group

$$\mathcal{D}_4 = \Big\{\big(\begin{smallmatrix}1&0\\0&1\end{smallmatrix}\big), \big(\begin{smallmatrix}0&-1\\1&0\end{smallmatrix}\big), \big(\begin{smallmatrix}-1&0\\0&-1\end{smallmatrix}\big), \big(\begin{smallmatrix}0&1\\-1&0\end{smallmatrix}\big), \big(\begin{smallmatrix}-1&0\\0&1\end{smallmatrix}\big), \big(\begin{smallmatrix}1&0\\0&-1\end{smallmatrix}\big), \big(\begin{smallmatrix}0&1\\1&0\end{smallmatrix}\big), \big(\begin{smallmatrix}0&-1\\-1&0\end{smallmatrix}\big)\Big\} \subset \mathcal{O}(2) \tag{6.9}$$

generated by the following elements of the two-dimensional orthogonal group $\mathcal{O}(2)$: four two-dimensional rotations by $\{0°, 90°, 180°, 270°\}$ and the two reflections with respect to the local coordinate axes, using the center pixel as origin. Since local grid coordinates are mapped onto each other, we can identify each transformation of the group $\mathcal{D}_4$ with a corresponding permutation σ of the pixel locations within the patch domain. Accordingly, writing with abuse of notation $\sigma \in \mathcal{D}_4$, the corresponding transformed patch (6.6) is given and denoted by

$$T_\sigma P_i = \big(f_{\sigma(k_1)}, \dots, f_i, \dots, f_{\sigma(k_{n_{\mathcal{P}}})}\big) \qquad k_1, \dots, k_{n_{\mathcal{P}}} \in \hat{\mathcal{N}}_{\mathcal{P},i}, \qquad \sigma \in \mathcal{D}_4. \tag{6.10}$$

We point out that no interpolation is required to compute these patch transformations.

In addition to the transformations (6.10), we consider all translations $P_i \mapsto P_k$, $k \in \hat{\mathcal{N}}_{\mathcal{P},i}$ of patch P_i mapping the center location i to some grid location k within its own region $\hat{\mathcal{N}}_{\mathcal{P},i} := \mathcal{N}_{\mathcal{P},i} \cap \hat{\mathcal{I}}$ restricted to interior pixels. We factor out these $|\mathcal{D}_4| \cdot n_{\mathcal{P}} = 8 \cdot n_{\mathcal{P}}$ degrees of freedom by considering all corresponding transformations of patch P_i as *equivalent*. These equivalence classes of patches provide the basis for invariant patch distances as defined next.

We define the *asymmetric patch distance* between two patches centered at pixel $i \in \hat{\mathcal{I}}$ and $k \in \hat{\mathcal{I}}$ by

$$d_{\mathcal{F}}(P_i, P_k) = \min_{\substack{\sigma \in \mathcal{D}_4 \\ j \in \hat{\mathcal{N}}_{\mathcal{P},i}}} \sum_{m \in [n_{\mathcal{P}}]} d_{\mathcal{F}}\big((T_\sigma P_j)_m, P_{k;m}\big) \tag{6.11}$$

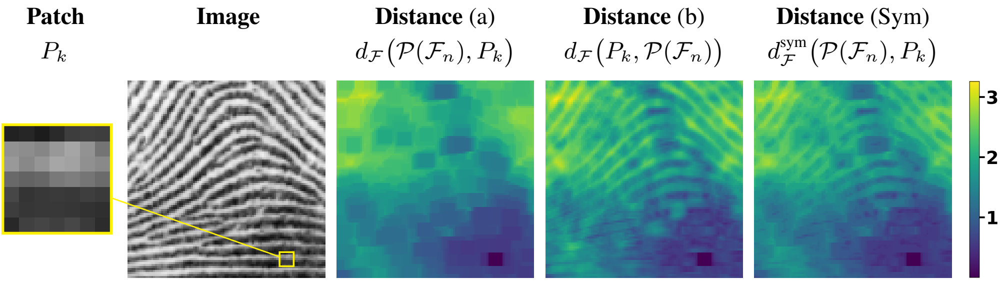


FIGURE 6.7. Visualization of the distance functions (6.11) and (6.12)) evaluated for a single patch P_k and all patches $\mathcal{P}(\mathcal{F}_n)$ of size $n_{\mathcal{P}} = 7 \times 7$ of the depicted image. The evaluation of distance (a) amounts to determine the minimal distance of P_k to *all* equivalence classes of patches generated by the patches of the entire image. As a consequence, equivalence classes close to P_k generate the 'blocky' graph of the distance function. Conversely, evaluation of distance (b) amounts to compare the *single* equivalence class generated by P_k to all image patches. As a consequence, the graph of the distance function reflects the original image structure in more detail. The symmetric distance (rightmost panel) is the pointwise minimum of distance (a) and (b). It is apparent that neither distance (a) nor (b) dominates the other distance.

and the *symmetric patch distance* by

$$d^{\text{sym}}_{\mathcal{F}}(P_i, P_k) = \min\big\{d_{\mathcal{F}}(P_i, P_k), d_{\mathcal{F}}(P_k, P_i)\big\}. \tag{6.12}$$

Figure 6.7 illustrates these locally invariant distance functions.

6.4.2. *Recovery of Patch Prototypes and Images.* Distance (6.12) defines the affinity matrix (3.2) by (3.3) and in turn the likelihood map (4.10) and the similarity map (2.35). As a consequence, the self-assignment flow can be integrated to obtain the assignment $W(t)$. We focus in this section on the recovery of prototypical patches and on 'explanations' of input images by assigning these prototypical patches. The corresponding results are illustrated by numerical examples in the subsequent Sections 6.4.3 and 6.4.4.

According to Section 3.3.2, prototypical patches representing each cluster are determined as weighted averages

$$P^*_j = \operatorname*{arg\,min}_{P \in \mathcal{P}(\mathcal{F})} \sum_{i \in \hat{\mathcal{I}}} \big(C(W)^{-1} W^{\top}\big)_{j,i} d^2_{\mathcal{F}}(P_i, P), \qquad j \in \mathcal{J}, \tag{6.13}$$

with respect to the *asymmetric* patch distance (6.11), since the prototypical patch $P \in \mathcal{P}(\mathcal{F})$ is not contained in the set of all image patches $\mathcal{P}(\mathcal{F}_n)$ (6.8).

Using these prototypes, the corresponding image is computed as follows. For each prototypical patch P^*_j, the optimal transformation for the assignment to pixel i is determined as

$$(\sigma^*_{i,j}, l^*_{i,j}) = \operatorname*{arg\,min}_{\substack{\sigma \in \mathcal{D}_4 \\ l \in \hat{\mathcal{N}}_{\mathcal{P},i}}} \sum_{m \in [n_{\mathcal{P}}]} d_{\mathcal{F}}\big((T_\sigma P_l)_m, P^*_{j;m}\big). \tag{6.14}$$

Using these transformations, a prototypical patch is assigned to every pixel $i \in \hat{\mathcal{I}}$. This implies that, for each pixel i, patches assigned to pixels $j \in \mathcal{N}_{\mathcal{P};i}$ may assign a corresponding patch entry to pixel i. Averaging these entries, normalized by the number of values contributed to pixel i, defines the restored image value at pixel i.

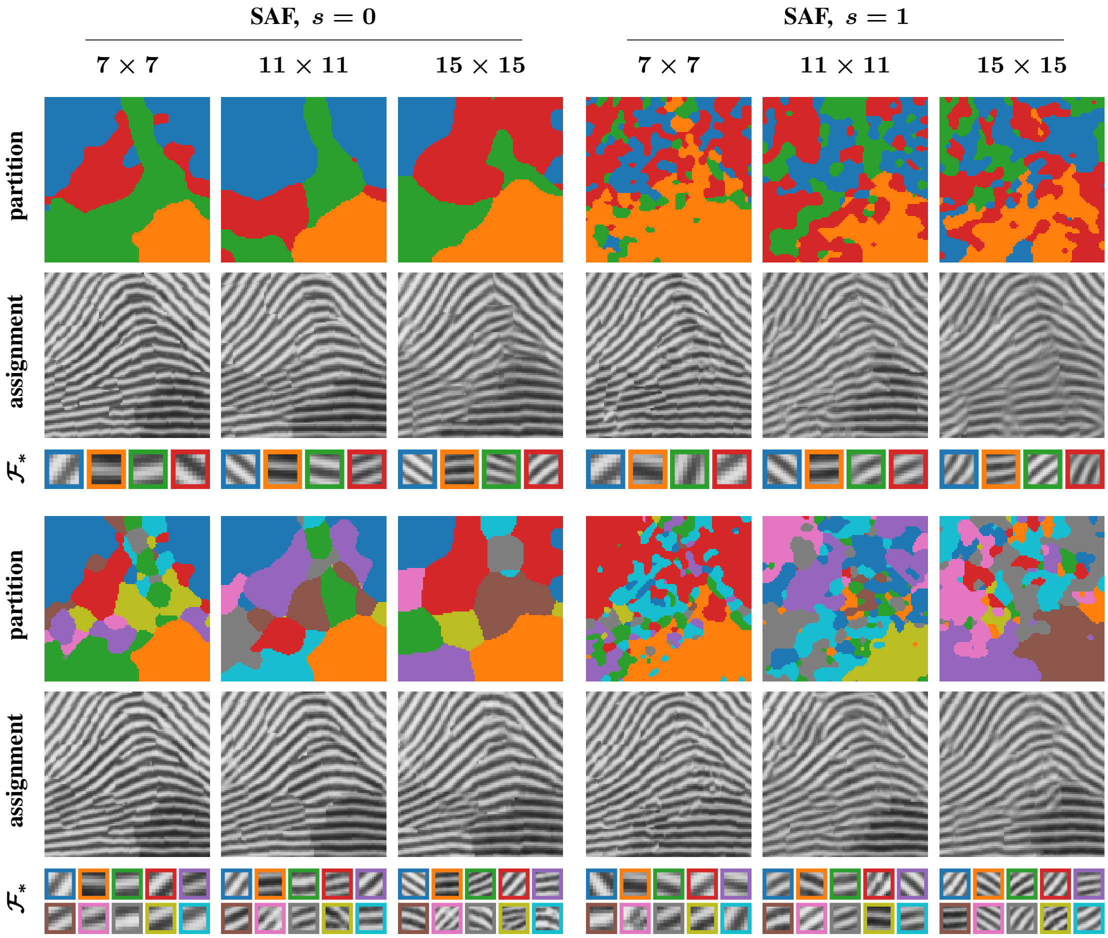


FIGURE 6.8. Determination of locally invariant patch prototypes, their assignment to the original image data and the corresponding partitions (depicted with pseudocolors), using the SAF ($s = 0$ and $s = 1$), different patch sizes (7×7, 11×11, 15×15) and numbers of prototypes ($c = 4$ and $c = 10$). The underlying transformation group enables accurate image representations even with $c = 4$ patches only, provided the patch size is close to the spatial scale of local image structure (here: 7×7 pixels). This performance deteriorates for larger patch sizes. The SAF with $s = 0$ yields partitions that are spatially more regular than the partitions computed with $s = 1$, since the latter tend to cover the feature space more uniformly, in agreement with the result depicted by by Figure 6.2.

6.4.3. *Patch-Based Self-Assignment Flow.* Figure 6.8 illustrates image partitions, the corresponding $c = 4$ and $c = 10$ prototypical patches of sizes $n_{\mathcal{P}} \in \{7 \times 7, 11 \times 11, 15 \times 15\}$, their assignment to the input image data as described in the preceding section, based on integrating the SAF with $s = 0$ and $s = 1$ and spatial regularization parameter $|\mathcal{N}| = 3 \times 3$.

In agreement with the discussion of the results depicted by Figure 6.2, we observe that the SAF with $s = 0$ returns partitions with a more regular spatial structure, whereas the SAF with $s = 1$ tends to cover the feature space more uniformly which is achieved with partitions that have a irregular spatial structure.

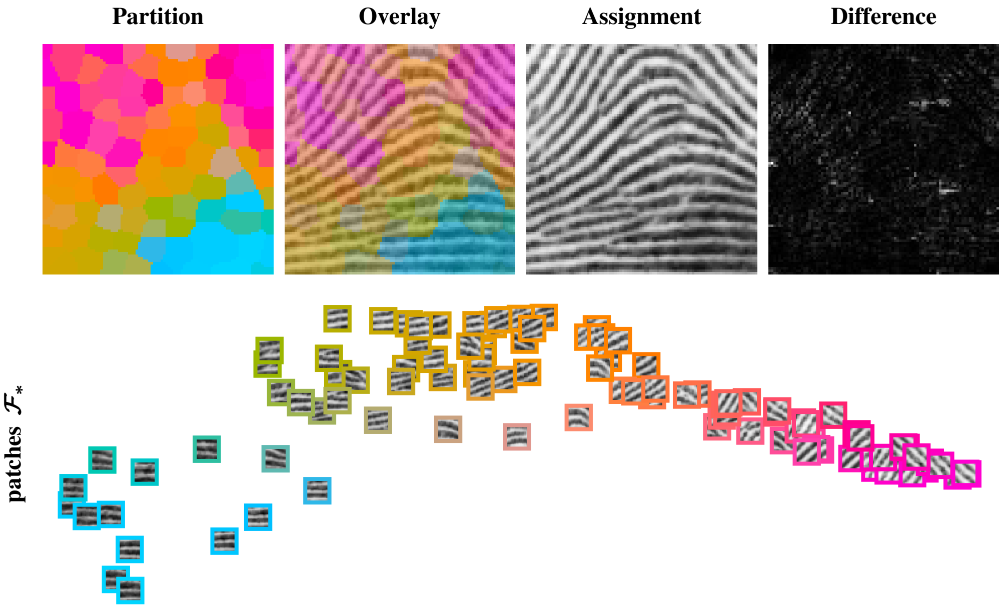


FIGURE 6.9. Experiment of Fig. 6.8 repeated with a larger patch dictionary leads to a detailed representation of local image structure. Although overlapping regions of assigned prototypical patches are averaged at each pixel in order to restore an image, the result 'Assignment' is quite close to the input data 'Image' of Fig. 6.7, due to using the locally invariant patch distance. Panel 'Difference' shows the difference as grayvalue plot (range $[0, 0.3]$). The lower panel displays a 2D embedding of the learned prototypical patches. The corresponding colors indicate their assignment in 'Partition' and 'Overlay'. Clusters in the lower panel, e.g. those colored pink and blue, illustrate the invariance under discrete rotations and reflections.

The image recovered by assigning the prototypical patches exhibits relatively sharp spatial structures, despite the small number of prototypes ($c \in \{4, 10\}$) and the pixel-wise averaging of grayvalues assigned by multiple patches. This illustrates that the small transformation group defined in Section 6.4.1 that does not even require image interpolation, actually is quite powerful. For example, the large blue region of the partition shown in Figure 6.8 that results from the SAF with $s = 0$ and 7×7 patches, indicates the optimal assignment of patches from a *single* equivalence class only. These patches fit quite accurately to image structures with different orientations and local edge profiles. This effect deteriorates when using patch sizes that are much larger than the typical variations of local image structure, as a comparison of the results for the patch size 15×15 with $c = 4$ and $c = 10$ shows.

For comparison, Figure 6.9 shows the result for a larger number $c = 100$ of prototypes, which leads to a detailed representation of local image structure. The lower panel displays a two-dimensional embedding of the weighted graph with prototypes as patches and the similarities (3.3) as weights. Representatives of equivalence classes of patches that are close to each other, are grouped together. Factoring out the group of transformations effectively copes with different edge profiles and orientations. Panel 'Difference' shows the absolute difference between the input image and labeling, ranging from 0 (black) to 0.3 (white).

6.4.4. *Patch Assignment to Novel Data.* We repeated the experiment illustrated by Figure 6.8 using the data shown in Figure 6.10. $c = 20$ locally invariant prototypical patches of size 7×7 pixels were

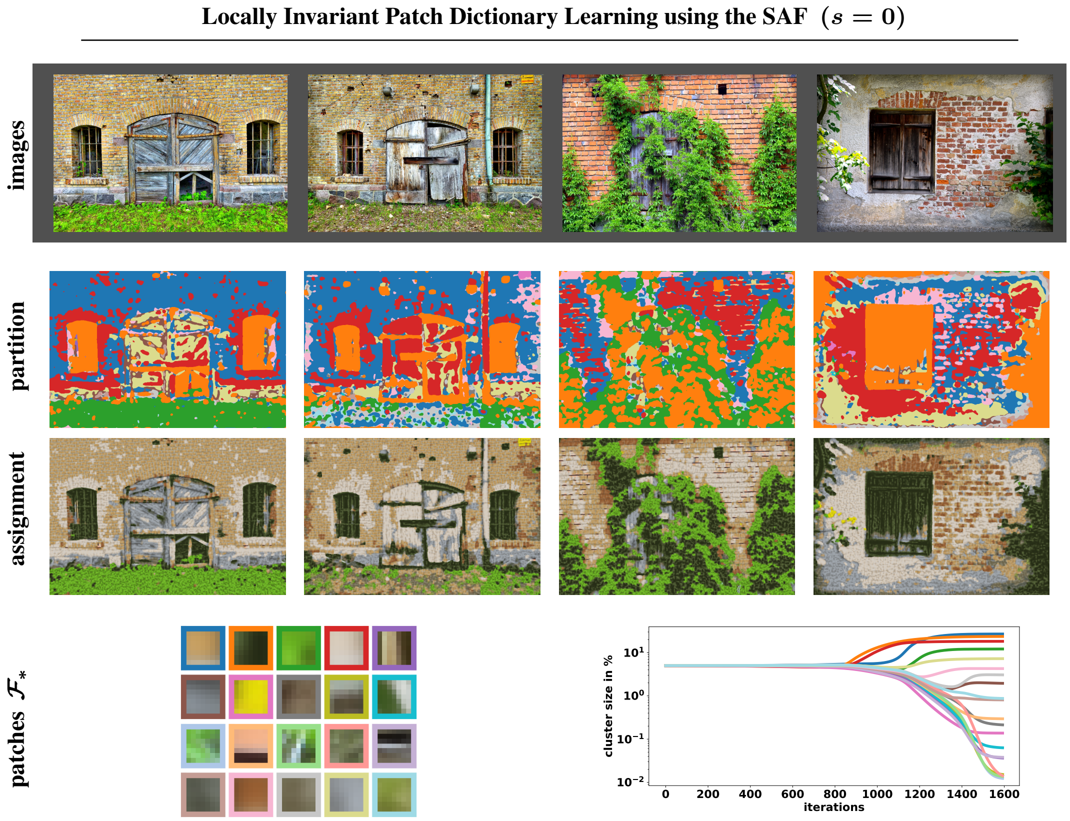


FIGURE 6.10. The bottom row shows a dictionary of $c = 20$ locally invariant patches of size 7×7 pixels, learned from the four images shown in the top row using the SAF with $s = 0$ and $|\mathcal{N}| = 3 \times 3$ pixels. The second and third row illustrate the patch assignments with pseudo-colors and the recovered image data, respectively. Closeness of the restored images to the input data, despite the small size of the patch dictionary, demonstrates the effectiveness of the underlying discrete transformation group. The evolution of cluster sizes (bottom row, right panel) illustrates the ability of the SAF to resolve 'conflicting' assignments due to mutually overlapping patches successfully, along with the formation of invariant patch prototypes, in a completely unsupervised way.

learned from 4 images using the SAF with $s = 0$ and $|\mathcal{N}| = 3 \times 3$ pixels. The restored images shown in the third row are remarkably close to the input data (first row), despite the small size $c = 20$ of the patch dictionary. This demonstrates again the effectiveness of the underlying discrete transformation group.

Figure 6.11 shows in the top row *novel* image data. These four images that are semantically similar to the training images of Figure 6.10 regarding the local image structure and texture (brick/stone, door/window, grass/ivy). The corresponding partitions and recovered images solely resulted from assigning the patch dictionary depicted by Figure 6.10 to the data by the supervised assignment flow. Again, the quality of image represention using this small dictionary is remarkable, except for the stone wall texture shown in column (c) of Figure 6.11, that is not present in the training data depicted by Figure 6.10.

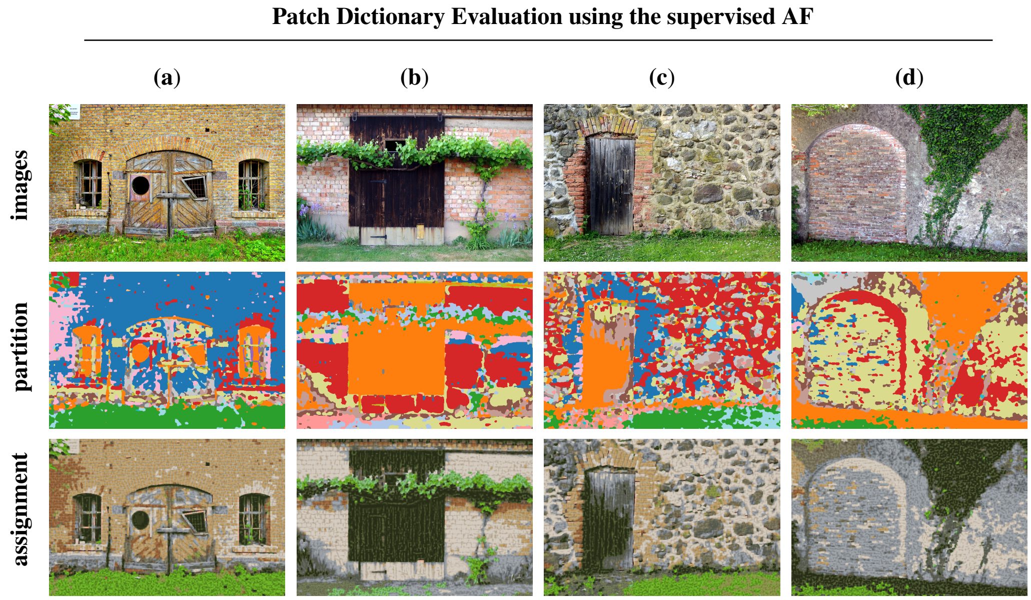


FIGURE 6.11. Supervised regularized assignment of the locally invariant patch dictionary from Figure 6.10 using the AF, to four *novel* images (top row). Since these images are semantically similar to the training data from Figure 6.10, the restored images are close to the input data, except for image (c) whose stone wall texture is not present in the training data.

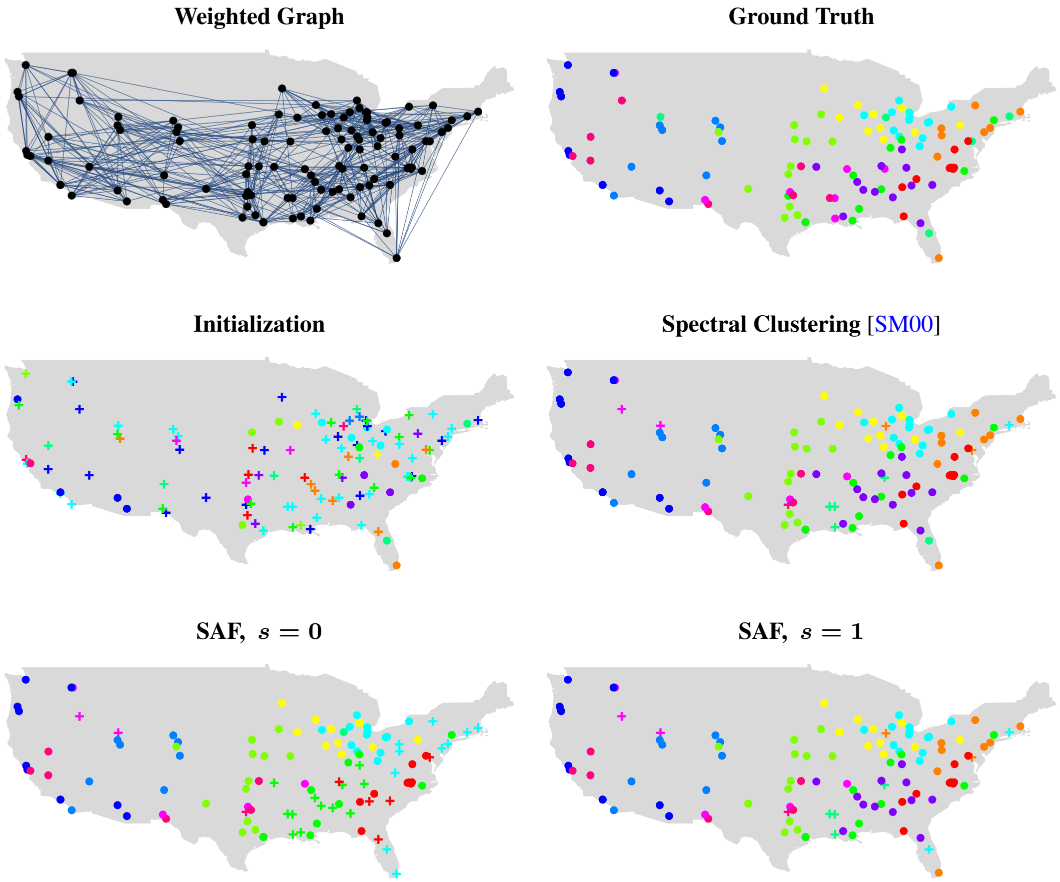
6.5. **Regularized Clustering of Weighted Graph Data.** Our approach can be applied to any data given on any undirected weighted graph. For illustration, we included an additional experiment using data not related to image analysis.

Figure 6.12 shows data in terms of a weighted graph $(\mathcal{I}, \mathcal{E}, K_{\mathcal{E}})$ adopted from [GN02]. It represents the network of American football games between Division IA colleges during the regular season fall 2000. Teams are subdivided into 12 conferences, mainly based on the geographical distance, that primarily play against each other in a first period. Afterwards, the conference champions play against each other in the final games. Each node of the network represents a team. Edge weights $K_{\mathcal{E}i,k}$ represent the number of games played between two teams. Labels for each vertex indicate the conference to which a team belongs, displayed by a corresponding color in Figure 6.12 (ground truth). We considered this labeling as ground truth for the task to partition the graph into $c = 12$ classes. The initial perturbation of the barycenter (4.11a) in terms of a distance matrix $D_{\mathcal{F},0}$ was computed by assigning feature vectors to each node based on the c dominant eigenvectors of $K_{\mathcal{F}}$, followed by greedy k-center clustering (Section 2.6). Markers ✚ indicate nodes that were assigned to a conference different from ground truth. Weights were defined as

$$w_{i,k} = \frac{\tilde{w}_{i,k}}{\langle \tilde{w}_i, \mathbb{1}_n \rangle} \quad \text{with} \quad \tilde{w}_{i,k} = K_{\mathcal{E}i,k} + \mathrm{Diag}(K_{\mathcal{E}} \mathbb{1}_n), \tag{6.15}$$

i.e. by adding the total number of games played by each team to the diagonal.

The nearest neighbor assignment of the initial distance matrix contains many erroneous assignments (Figure 6.12, initialization). The results of the SAF with $s = 1$ reproduces almost the ground-truth labeling and is also close to the result of applying spectral clustering [SM00] directly to $K_{\mathcal{E}}$. The


Weighted Graph

Ground Truth

Initialization

Spectral Clustering [SM00]

SAF, $s = 0$

SAF, $s = 1$

FIGURE 6.12. Weighted graph data of American football games between Division IA colleges during the regular season fall 2000 are clustered. Each node represents a team and edge weights indicate the number of games played between two teams. The colored nodes in 'Ground Truth' show the subdivision of the teams into 12 conferences (clusters), that primarily play against each other in a first period. Graph partitioning with $c = 12$ was performed using the SAF with $s = 0$ and $s = 1$, and with weights defined by (6.15). Markers ✚ indicate labels assigned to nodes that differ from ground truth. Starting from the initialization (2nd row, left panel) which is noisy, the SAF with $s = 1$ returns almost the ground-truth labeling and is also close to the result of directly applying spectral clustering to $K_{\mathcal{E}}$. The SAF with $s = 0$ enforces label assignments that are spatially more regular, and with empty clusters orange and purple.

SAF with $s = 0$ enforces assignments with a more regular spatial structure. Both findings agree with observations made in preceding experiments; see e.g. Figure 6.2.

7. Conclusion

We extended the assignment flow approach to supervised image labeling introduced by [ÅPSS17] to unsupervised scenarios where no labels are available. The resulting self-assignment flow takes a pairwise affinity matrix as input data and maximizes the correlation (inner product) with a low-rank self-assignment matrix, corresponding to a factorization with the variables of the assignment flow. A single parameter $s \in [0,1]$ determines the self-assignment matrix as smooth geodesic interpolation of the self-affinity matrix ($s = 0$) and the self-influence matrix ($s = 1$), which enables to control the relative influence of spatial regularization and the preservation of feature-induced local image structure, respectively. A second parameter, the size $|\mathcal{N}|$ of local neighborhoods for geometric averaging of assignments, controls the scale of the resulting image partition as in the supervised case.

The compositional design of the approach, informally expressed as 'regularization $\circ$ data likelihood' as opposed to 'regularization $+$ data likelihood' as in traditional variational approaches, merely required to generalize the likelihood map (cf. (4.10)) in order to extend the approach to the unsupervised case. In particular, numerical techniques developed by [ZSPS19] for integrating the assignment flow still apply. Learning patch dictionaries with a locally invariant patch distance function demonstrated exemplarily, together with a range of further numerical experiments, that our approach can flexibly cope with all common feature representations, including RKHS embeddings.

We characterized mathematically our approach from different relevant viewpoints, depending on the parameter s: As rank-constrained discrete optimal transport and as normalized spectral cuts that are spatially regularized in an unbiased way (rather than adding spatial coordinates as 'features'). Additionally, we showed that the formation of prototypes automatically optimizes a classical class separability measure. Finally, from the viewpoint of combinatorial optimization, our approach successfully handles completely positive factorizations of self-assignments in large-scale scenarios, subject to spatial regularization.

Promising directions of further research include application-dependent extensions of the invariance group in order to learn compact patch dictionaries using the self-assignment flow in various scenarios. An open challenging problem concerns the extension of weight parameter estimation for application-specific adaptive regularization [HSPS19] to the unsupervised self-assignment flow approach.

Acknowledgements

Financial support by the German Science Foundation (DFG), grant GRK 1653, is gratefully acknowledged. This work has also been stimulated by the Heidelberg Excellence Cluster STRUCTURES, funded by the DFG under Germanys Excellence Strategy EXC-2181/1 - 390900948.

References

[AMS09] P.-A. Absil, R. Mahony, and R. Sepulchre, *Optimization Algorithms on Matrix Manifolds*, Princeton University Press, 2009.

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

[And79] T. Ando, *Generalized Schur Complements*, Lin. Algebra Appl. **27** (1979), 173–186.

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

[BCPD99] R. E. Burkhard, E. Cela, P. M. Pardalos, and D. Z. Du, *Linear Assignment Problems and Extensions*, pp. 75–149, Kluwer Acad. Publ., 1999.

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

[Bom18] I. M. Bomze, *Building a Completely Positive Factorization*, Central Europ. J. Oper. Res. **26** (2018), no. 2, 287–305.

[BP94] A. Berman and R. J. Plemmons, *Nonnegative Matrices in the Mathematical Sciences*, SIAM, 1994.

[DK82] P. A. Devijver and J. Kittler, *Pattern Recognition: A Statistical Approach*, Prentice Hall, 1982.

[DM05] P. Drineas and M. W. Mahoney, *On the Nyström Method for Approximating a Gram Matrix for Improved Kernel-Based Learning*, J. Mach. Learning Res. **6** (2005), 2153–2175.

[GM16] A. Gittens and M. W. Mahoney, *Revisiting the Nyström Method for Improved Large-Scale Machine Learning*, J. Mach. Learning Res. **17** (2016), no. 1, 3977–4041.

[GN02] M. Girvan and M. E.-J. Newman, *Community Structure in Social and Biological Networks*, Proceedings of the National Academy of Sciences **99** (2002), no. 12, 7821–7826.

[HM96] U. Helmke and J. B. Moore, *Optimization and Dynamical Systems*, 2nd ed., Springer, 1996.

[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*, CoRR abs/1910.09976 (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.

[KYP15] D. Kuang, Sa. Yun, and H. Park, *SymNMF: Nonnegative Low-Rank Approximation of a Similarity Matrix for Graph Clustering*, Journal of Global Optimization **62** (2015), no. 3, 545–574.

[Lau87] S. L. Lauritzen, *Chapter 4: Statistical Manifolds*, Differential Geometry in Statistical Inference (Shanti S. Gupta, S. I. Amari, O. E. Barndorff-Nielsen, R. E. Kass, S. L. Lauritzen, and C. R. Rao, eds.), Institute of Mathematical Statistics, Hayward, CA, 1987, pp. 163–216.

[Pey18] M. Peyré, G.and Cuturi, *Computational Optimal Transport*, CNRS, 2018.

[RW95] F. Rendl and H. Wolkowicz, *A Projection Technique for Partitioning the Nodes of a Graph*, Ann. Operations Res. **58** (1995), 155–179.

[San15] F. Santambrogio, *Optimal Transport for Applied Mathematicians*, Birkhäuser, 2015.

[Sch19] C. Schnörr, *Assignment Flows*, Variational Methods for Nonlinear Geometric Data and Applications (P. Grohs, M. Holler, and A. Weinmann, eds.), Springer (in press), 2019.

[SM00] J. Shi and J. Malik, *Normalized Cuts and Image Segmentation*, IEEE Transactions on Pattern Analysis and Machine Intelligence **22** (2000), no. 8, 888–905.

[SW14] M. Storath and A. Weinmann, *Fast Partitioning of Vector-Valued Images*, SIAM Journal on Imaging Sciences **7** (2014), no. 3, 1826–1852.

[Vil09] C. Villani, *Optimal Transport: Old and New*, Springer, 2009.

[vL07] U. von Luxburg, *A Tutorial on Spectral Clustering*, Statistics and Computing **17** (2007), no. 4, 395–416.

[WS01] C. K. I. Williams and M. Seeger, *Using the Nyström Method to Speed up Kernel Machines*, Proc. NIPS, 2001, pp. 682–688.

[YC16] Z. Yang and E. Corander, J.and Oja, *Low-Rank Doubly Stochastic Matrix Decomposition for Cluster Analysis*, Journal of Machine Learning Research **17** (2016), no. 1, 6454–6478.

[ZS05] R. Zass and A. Shashua, *A Unifying Approach to Hard and Probabilistic Clustering*, Proc. ICCV, 2005.

[ZSPS19] A. Zeilmann, F. Savarino, S. Petra, and C. Schnörr, *Geometric Numerical Integration of the Assignment Flow*, CoRR abs/1810.06970, Inverse Problems: in press (2019).

[ZZPS19a] A. Zern, M. Zisler, S. Petra, and C. Schnörr, *Unsupervised Assignment Flow: Label Learning on Feature Manifolds by Spatially Regularized Geometric Assignment*, CoRR abs/1904.10863 (2019).

[ZZPS19b] M. Zisler, A. Zern, S. Petra, and C. Schnörr, *Unsupervised Labeling by Geometric and Spatially Regularized Self-Assignment*, Proc. SSVM, Springer, 2019.

(M. Zisler) Image and Pattern Analysis Group, Heidelberg University, Germany
E-mail address: `zisler@math.uni-heidelberg.de`

(A. Zern) Image and Pattern Analysis Group, Heidelberg University, Germany
E-mail address: `artjom.zern@iwr.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`