I N A U G U R A L – D I S S E R T A T I O N

zur

Erlangung der Doktorwürde

der

Naturwissenschaftlich–Mathematischen Gesamtfakultät

der

R U P R E C H T – K A R L S – U N I V E R S I T Ä T

H E I D E L B E R G

vorgelegt von

M.Sc. Ruben Hühnerbein

aus Göttingen

Tag der mündlichen Prüfung

15. April 2020

Inference and Model Parameter Learning for Image Labeling by Geometric Assignment

Advisor

Prof. Dr. Christoph Schnörr

Prof. Dr. Ekaterina Kostina

Zusammenfassung

Die Bildsegmentierung ist ein grundlegendes Problem im Bereich der Bildverarbeitung. In dieser Arbeit präsentieren wir jeweils einen neuen Ansatz zur Maximum A Posteriori (MAP) Inferenz und zum Lernen der Modellparameter für die Bildsegmentierung. Beide Ansätze werden in einem glatten geometrischen Rahmen formuliert, deren jeweiliger Lösungsraum eine einfache Riemannsche Mannigfaltigkeit ist. Numerische Optimierungsschritte bestehen aus multiplikativen Updates, die den resultierenden Riemannschen Gradientenfluss geometrisch integrieren.

Unser neuer Ansatz zur MAP-Inferenz basiert auf diskreten graphischen Modellen. Mittels lokaler Wasserstein Distanzen koppeln wir an jeder Kante die Zuordnungsmaße des zugrunde liegenden Graphen. Dadurch wird die gegebene diskrete Zielfunktion glatt approximiert und auf die Zuordnungs-Mannigfaltigkeit beschränkt. Ein entsprechendes Diskretisierungsschema kombiniert die geometrische Integration des resultierenden Gradientenflusses mit der Rundung zu integralen Lösungen, die zulässige Segmentierungen darstellen. Diese Formulierung stellt eine innere Relaxierung des diskreten Segmentierungsproblems dar, bei der die lokalen Marginalisierungsnebenbedingungen, die aus der etablierten Relaxierung der linearen Programmierung bekannt sind, jederzeit erfüllt sind.

Desweiteren untersuchen wir das inverse Problem des Modellparameter-Lernens unter Verwendung des linearen Zuordnungsflusses und Trainingsdaten, bei denen die Segmentierung bekannt ist. Dies wird durch einen Riemannschen Gradientenfluss auf der Mannigfaltigkeit der Parameter, welche die Regularisierungseigenschaften des Zuordnungsflusses bestimmen, erreicht. Diese glatte dynamische Formulierung ermöglicht es das Problem des Modellparameter-Lernens aus der Perspektive der Parameterschätzung dynamischer Systeme anzugehen. Mit Hilfe von symplektisch partitionierten Runge–Kutta Methoden zur numerischen Integration wird gezeigt, dass die Herleitung der Sensitivitätsbedingungen des Parameter-Lernproblems und dessen Diskretisierung kommutieren. Eine wichtige Konsistenzeigenschaft unseres Ansatzes ist, dass das Lernen auf exakter Inferenz basiert.

Abstract

Image labeling is a fundamental problem in the area of low-level image analysis. In this work, we present novel approaches to maximum a posteriori (MAP) inference and model parameter learning for image labeling, respectively. Both approaches are formulated in a smooth geometric setting, whose respective solution space is a simple Riemannian manifold. Optimization consists of multiplicative updates that geometrically integrate the resulting Riemannian gradient flow.

Our novel approach to MAP inference is based on discrete graphical models. By utilizing local Wasserstein distances for coupling assignment measures across edges of the underlying graph, we smoothly approximate a given discrete objective function and restrict it to the assignment manifold. A corresponding update scheme combines geometric integration of the resulting gradient flow, and rounding to integral solutions that represent valid labelings. This formulation constitutes an inner relaxation of the discrete labeling problem, i.e. throughout this process local marginalization constraints known from the established linear programming relaxation are satisfied.

Furthermore, we study the inverse problem of model parameter learning using the linear assignment flow and training data with ground truth. This is accomplished by a Riemannian gradient flow on the manifold of parameters that determine the regularization properties of the assignment flow. This smooth formulation enables us to tackle the model parameter learning problem from the perspective of parameter estimation of dynamical systems. By using symplectic partitioned Runge–Kutta methods for numerical integration, we show that deriving the sensitivity conditions of the parameter learning problem and its discretization commute. A favorable property of our approach is that learning is based on exact inference.

Acknowledgements

At this point, I would like to thank everyone who supported me during the creation of this work. First of all, I would like to thank my advisor *Prof. Dr. Christoph Schnörr* for his excellent support during my Ph.D. studies. His work ethic and accessibility are to be emphasized: I could always knock on his door and he gave me the necessary new input or a different point of view. This is not a matter of course and has helped to organize my work and to combine family and working life.

For the pleasant and cooperative working atmosphere of the past years I would like to thank all my colleagues of the *Image & Pattern Analysis Group*: *Fabrizio Savarino, Artjom Zern, Matthias Zisler, Jan Plier, Alexander Zeilmann, Dmitrij Sitenko, Lukas Kiefer, Bastian Boll, Jonathan Schwarz* and *Felix Draxler*. I especially like to thank my outstanding office colleague *Fabrizio* for all discussions and conversations on private or professional topics, thousands of coffees and joint working trips. For the support in organizational matters my thanks go to *Evelyn Wilhelm*. I am especially grateful for her optimistic and happy spirit she spreads within the group. I also thank *Barbara Werner* for helping me with bureaucratic matters.

I would like to thank *Dr. Jürgen Gutekunst, Johannes Herold* and *Dennis Röhner* for proofreading parts of this work. For their support in doctoral matters I would like to thank *Dorothea Heukäufer* and *Dr. Rebecca Paimann* from the Dean's Office of the Faculty of Mathematics and Computer Science.

Funding by the Deutsche Forschungsgesellschaft via the research training group 1653 *Spatio/Temporal Graphical Models and Applications in Image Analysis* is also gratefully acknowledged. In addition, part of this research was done while I was allowed to visit the Institute for Pure and Applied Mathematics (IPAM) at UCLA, Los Angeles. This institute is supported by the National Science Foundation (Grant No. DMS-1440415), for whose financial assistance I am grateful.

I thank my little son *Joris* for the fact that he doesn't care what I do for a living and he is just happy when his parents are at home and spent time with him.

Finally, I would like to thank my wife *Ellen* for being at my side for nine years now, for her endless support and encouragement, and that we live our dreams together.

Heidelberg, February 2020 *Ruben Hühnerbein*

x

Contents

List of Publications XV

1 Introduction 1

1.1 Motivation . . . 1
1.1.1 Inference . . . 2
1.1.2 Learning . . . 3
1.2 Related Work . . . 4
1.2.1 Image Labeling . . . 4
1.2.2 Parameter Estimation for Dynamical Systems . . . 6
1.3 Outline and Contribution . . . 7
1.4 Notation . . . 9

2 Mathematical Background 13

2.1 Elements of Differential Geometry . . . 13
2.1.1 Vector Fields and Integral Curves . . . 14
2.1.2 Connections . . . 15
2.1.3 Riemannian Geometry . . . 17
2.2 Probabilistic Graphical Models . . . 18
2.2.1 Discrete Graphical Models . . . 18
2.2.2 Inference . . . 19
2.2.3 Learning . . . 20
2.2.4 Linear Programming Relaxation . . . 21
2.2.5 Loopy Belief Propagation . . . 23
2.3 Parameter Estimation of Dynamical Systems . . . 24
2.3.1 Sensitivity Analysis . . . 25
2.3.2 Symplectic Partitioned Runge–Kutta Methods . . . 27
2.3.3 Computing Adjoint Sensitivities . . . 29
2.3.4 Adjoint Sensitivity: Two Specific Numerical Schemes . . . 33

3 Image Labeling by Assignment 37

3.1 The Assignment Manifold . . . 37
3.1.1 Local Object: Relative Interior of the Probability Simplex . . . 37

3.1.2 Global Object: Assignment Manifold . . . 43
3.2 Vector Fields on the Assignment Manifold . . . 45
3.3 Image Labeling on the Assignment Manifold . . . 46
3.3.1 Assignment Flow . . . 47
3.3.2 Numerical Integration of the Flow . . . 48
3.4 Experiments . . . 51
3.4.1 Implementation Details . . . 51
3.4.2 Parameter Influence . . . 52
3.5 Extensions . . . 52

4 Inference based on Graphical Models and Assignment 57
4.1 Objective Function . . . 57
4.1.1 Smooth Approximation of the LP Relaxation . . . 58
4.2 Global Euclidean Gradient . . . 61
4.3 Local Wasserstein Gradient . . . 62
4.3.1 Formula of the Local Wasserstein Gradient . . . 62
4.3.2 Computation of the Local Wasserstein Gradient . . . 65
4.4 Graphical Models on the Assignment Manifold . . . 71
4.4.1 Combination of Minimizing and Rounding . . . 71
4.4.2 Connection to Belief Propagation . . . 73
4.5 Experiments . . . 75
4.5.1 Parameter Influence . . . 76
4.5.2 Cyclic Graphical Models on $\mathcal{K}^3$ . . . 80
4.5.3 Comparison to Other Methods . . . 85
4.5.4 Non-Potts Prior . . . 86

5 Model Parameter Learning for Adaptive Regularization 91
5.1 Problem Formulation . . . 91
5.1.1 The Parameter Manifold . . . 92
5.1.2 Objective Function . . . 93
5.1.3 Modified Linear Assignment Flow . . . 94
5.2 Gradients and Differentials . . . 94
5.3 Numerical Optimization . . . 97
5.4 Experiments . . . 98
5.4.1 Adaptive Regularization of Curvilinear Line Structures . . . 99
5.4.2 Pattern Formation by Label Transport . . . 105

6 Conclusion **113**

A Proofs **117**

A.1 Proofs of Chapter 2 117

A.2 Proofs of Chapter 3 125

A.3 Proofs of Chapter 4 128

A.4 Proofs of Chapter 5 136

B Derivation of Loopy Belief Propagation **139**

Bibliography **145**

Nomenclature **153**

Figures, Tables & Acronyms **155**

List of Publications

Partial results of this thesis have been published in the following papers:

1. Åström, F.; Hühnerbein, R.; Savarino, F.; Recknagel, J. and Schnörr, C. MAP Image Labeling Using Wasserstein Messages and Geometric Assignment. In *Scale Space and Variational Methods in Computer Vision (SSVM 2017)*, pages 373–385, 2017.

2. Savarino, F.; Hühnerbein, R.; Åström, F.; Recknagel, J. and Schnörr, C. Numerical Integration of Riemannian Gradient Flows for Image Labeling. In *Scale Space and Variational Methods in Computer Vision (SSVM 2017)*, pages 361–372, 2017.

3. Hühnerbein, R.; Savarino, F.; Aström, F. and Schnörr, C. Image Labeling Based on Graphical Models Using Wasserstein Messages and Geometric Assignment. In *SIAM J. Imaging Science*, 11 (2): 1317-1362, 2018.

4. Hühnerbein, R.; Savarino, F.; Petra, S. and Schnörr, C. Learning Adaptive Regularization for Image Labeling Using Geometric Assignment. In *Scale Space and Variational Methods in Computer Vision (SSVM 2019)*, pages 393–405, 2019.

5. Hühnerbein, R.; Savarino, F.; Petra, S. and Schnörr, C. Learning Adaptive Regularization for Image Labeling Using Geometric Assignment. https://arxiv.org/pdf/1910.09976.pdf, 2019. preprint.

Chapter 1

Introduction

1.1 Motivation

Image labeling is a thoroughly investigated problem in the area of low-level image analysis. The task is to segment a given image into coherent regions, where all pixels belonging to the same region should share certain characteristics. This is an important step in the analysis of the image content and has many different practical applications, e.g. object detection, medical imaging or pattern recognition. Nevertheless, this thesis focuses on basic mathematical research for image analysis with no specific application in mind.

input image **labeling**

Figure 1.1: Image labeling example. LEFT: Exterior view of the *Alte Universität* in Heidelberg, Germany[a]. RIGHT: This image shows the labeling result consisting of 11 colors which served as labels.

[a] Außenansicht der Universität Heidelberg - © Achim Mende, https://m.tourismus-bw.de/Media/Attraktionen/Universitaet-Heidelberg.

For example, consider the image labeling scenario depicted in Fig. 1.1. We would like to assign a label of a predefined set (11 colors) to each pixel. The left image

shows the input image consisting of many different colors, whereas the right image demonstrates a labeling result consisting of 11 colors.

How do we judge whether a given labeling is good or bad? From a purely visual perspective, the labeling should be close to the input data and should have locally the same labels. *Locally* means that the label assignment should not jump back and forth in a local window of the labeling. From a mathematical perspective, the answer to this question depends on the chosen model. In any case these *visual objectives* should be taken into account when designing a suitable mathematical model.

Nearly all models have parameters that can be set differently in order to adjust the respective model to a given scenario. In case the parameters are given, the task is to *infer* a labeling from a given image. That is why this task is commonly called *inference*. In contrast, the second task is the other way around, i.e. the model parameters are unknown, and both an input image and a corresponding ground truth labeling are given. We then wish to *learn* the model parameters from this data. Therefore, this scenario in which both the image and a ground truth labeling are given is commonly called *supervised learning*. In order to specify the research objectives of this thesis, we introduce in the subsequent sections the tasks of inference and learning in mathematical terms.

1.1.1 Inference

A prominent model for tackling the inference task is the *discrete graphical model*. The model uses a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to represent the observed image data. Each node $i \in \mathcal{V}$ indexes a pixel location x_i which takes a value from a discrete set of *labels*

$$x_i \in \mathcal{X} = \{\ell_1, \dots, \ell_n\}. \tag{1.1}$$

Now, the image labeling problem is formulated as minimization problem of the *discrete energy function*

$$\min_{x \in \mathcal{X}^m} E(x), \qquad E(x) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{ij \in \mathcal{E}} \theta_{ij}(x_i, x_j), \tag{1.2}$$

where the variables θ_i and θ_{ij} denote the given model parameters. The energy function $E(x)$ has the format of variational problems comprising a *data term* and a *regularizer*. From a Bayesian perspective, therefore, minimizing E corresponds to *Maximum A Posteriori* inference with respect to the probability distribution $p(x) = \frac{1}{Z}\exp(-E(x))$.

Since (1.2) is a combinatorially hard problem, a major class of algorithms for approximately solving (1.2) is based on the *linear programming relaxation* (see Section 2.2.4 for details)

$$\min_{\mu \in \mathcal{L}_\mathcal{G}} \langle \theta, \mu \rangle. \tag{1.3}$$

The globally optimal solution of the linear program (LP) (1.3) is the *relaxed indicator vector* μ whose components take values in $[0,1]$. If μ is a binary vector, then it corresponds to a solution of the discrete problem (1.2). However, in realistic applications this is not the case, and the relaxed solution μ has to be *rounded* to an integral solution in a post-processing step.

In this thesis, we present an alternative inference algorithm that deviates from the traditional two-step process above: convex relaxation and rounding. It is based on the geometric approach [8] to image labeling. The basic idea of [8] is to follow vector fields on the the relative interior of the probability simplex, equipped with the Fisher-Rao metric, and to regularize label assignments by computing Riemannian means. This results in a parallel and multiplicative update scheme that converges to an integral solution (see Chapter 3 for details).

Adopting the smooth geometric approach [8], our *research objectives* are:

- Extend the approach [8] to efficiently compute a high-quality (low-energy) solution for an arbitrary given instance of the discrete labeling problem (1.2).
- Develop a novel inference algorithm which combines both relaxation and rounding to an integral solution in a *single process*.
- Overcome the inherent non-smoothness of convex relaxations by sticking to the smooth geometric model [8].

1.1.2 Learning

While the inference task of discrete graphical models (1.2) is well understood, the task of *learning* the model parameters of such models is less explored and has remained elusive. Especially for models with higher connectivity of the underlying graph the learning task becomes hard. In particular the *relation* between approximations of the *learning problem* on the one hand, and approximations of the underlying inference subproblem on the other hand, is less understood.

Based on the problem formulation of a general parameter estimation problem

$$\min_{p \in \mathcal{P}} \quad \mathcal{C}\big(x(T)\big) \tag{1.4a}$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), p, t), \quad t \in [0, T], \tag{1.4b}$$

$$x(0) = x_0, \tag{1.4c}$$

our *research objectives* are:

- Solve the *inference* (labeling) subroutine *exactly* by means of numerically solving (1.4b). The advantage is that errors of approximate inference as they occur with graphical models are absent and cannot compromise the effectiveness of parameter learning.

- Show that the operations of (i) deriving the adjoint sensitivity conditions of (1.4) and (ii) problem discretization *commute* if a proper numerical scheme is used.

- Design a prediction map that maps features extracted from *novel* data to proper weights as regularization parameters.

1.2 Related Work

1.2.1 Image Labeling

In typical applications of image labeling the problem sizes of the linear program (LP) (1.3) are too large to use standard LP codes. In particular, the theoretically and practically most efficient interior point methods based on self-concordant barrier functions [49, 57] are infeasible due to the dense linear algebra steps required to determine search and update directions.

Therefore, the need for dedicated solvers for the LP relaxation (1.3) has stimulated a lot of research. A prominent example is constituted by subclasses of objective functions (1.2) as studied in [43], in particular binary submodular functions, that allow for reformulating the labeling problem as maximum-flow problem in an associated network and the application of discrete combinatorial solvers [15, 14].

Since the structure of such algorithms inherently limits parallel implementations, *belief propagation* and variants [80] have been popular in the literature. These fixed point schemes in terms of dual variables iteratively enforce the so-called local polytope constraints that define the feasible set of the LP relaxation (1.3). They can be

efficiently implemented using *message passing* and exploit the structure of the underlying graph. Although convergence is not guaranteed on cyclic graphs, the performance in practice may be good [79]. These theoretical deficiencies of basic belief propagation stimulated research on *convergent* message passing schemes, either using heuristic damping or utilizing in a more principled way *convexity*. Prominent examples of the latter case are [75, 33]. We refer to [38] for many more references and a comprehensive experimental evaluation of a broad range of algorithms for image labeling.

The feasible set of the relaxation (1.3) is a superset of the original feasible set of (1.2). Therefore, globally optimal solutions to (1.3) generally do *not* constitute valid labelings but comprise *non*-integral components $\mu_i(x_i) \in (0,1)$, $x_i \in \mathcal{X}$, $i \in \mathcal{V}$. Randomized rounding schemes for converting a relaxed solution vector $\overline{\mu}$ to a valid labeling $x \in \mathcal{X}^m$, along with suboptimality bounds, were studied in [40, 19]. The problem of inferring components x_i^* of the unknown globally optimal *combinatorial* labeling that minimizes (1.2), through partial optimality and persistency, was studied in [71]. We refer to [77] for more information about the LP relaxation of labeling problems, and to [74] for connections to discrete probabilistic graphical models from the variational viewpoint.

The approach [56] applies the mirror descent scheme [48] to the LP (1.3). This amounts to sequential proximal minimization [60], yet using a Bregman distance as proximity measure instead of the squared Euclidean distance [18]. A key technical aspect concerns the proper choice of entropy functions related to the underlying graphical model, that qualify as convex functions of Legendre type (cf. [10]). The authors of [56] observed a fast convergence rate. However, the scheme does not scale up to the typically large problem sizes used in image analysis, especially when graphical models with higher edge connectivity are considered, due to the memory requirements when working entirely in the primal domain.

Optimal transport and the *Wasserstein distance* have become a major tool of signal modeling and analysis [44]. In connection with the metric labeling problem, using the Wasserstein distance (also known as optimal transport costs, earthmover metrics) was proposed before by [1] and [19]. These works study bounds on the integrality gap of an *earthmover LP* and performance guarantees of rounding procedures applied as post-processing. While the earthmover LP corresponds to our approach (4.7) *without* smoothing, the authors do not specify how to solve such LPs efficiently, especially when the LP relates to a large-scale graphical models as in image analysis. Moreover, the bounds derived by [1] become weak with increasing numbers of variables, which are fairly large in typical problems of image analysis. In contrast, the focus of the present paper is on a *smooth geometric* problem reformulation that scales

well with both the problem size and the number of labels, and performs rounding *simultaneously*. If and how theoretical guarantees regarding the integrality gap and rounding carry over to our setting, is an interesting open problem of future research.

Regarding the finite-dimensional formulation of optimal discrete transport in terms of linear programs, the design of efficient algorithms for large-scale problems requires sophisticated techniques [66]. The problems of discrete optimal transport studied in this thesis, in connection with the local Wasserstein distances of (4.7), have a small or moderate size (n^2: number of labels squared). We apply the standard device of enhancing convexity through entropic regularization, which increases smoothness in the dual domain. We refer to [68] and [16, Ch. 9] for basic related work and the connection to matrix scaling algorithms. If entropic regularization is very weak or the problem sizes are large, the related fixed point iteration suffers from numerical instability. Dedicated methods for handling these instabilities have been proposed [67]. Smoothing of the Wasserstein distance and Sinkhorn's algorithm has become popular in machine learning due to [20]. The authors of [52, 21] comprehensively investigated barycenters and interpolation based on the Wasserstein distance. Our approach to image labeling, in conjunction with the geometric approach of [8], is novel and elaborates on [7].

Since our approach is defined on a graph and works with data on a graph, our work may be assigned to the broad class of nonlocal methods for image analysis on graphs, from a more general viewpoint. Recent major related work includes [12] on the connection between the Ginzburg-Landau functional for binary regularized segmentation and spectral clustering, and [11] on generalizing PDE-like models on graphs to manifold-valued data. We refer to the bibliography in these works and to the seminal papers [4] on regularized variational segmentation using Γ-convergence and to [26, 25] on nonlocal variational image processing on graphs, that initiated these fast evolving lines of research. However, the focus of this thesis is on discrete graphical models and the corresponding labeling problem, in terms of any discrete objective function of the form (1.2).

1.2.2 Parameter Estimation for Dynamical Systems

The task of optimizing parameters of a dynamical system (1.4) is a familiar one in the communities of scientific computing and optimal control [73, 17], but may be less known to the imaging community.

Geometric numerical integration of ordinary differential equations (ODEs) on manifolds is a mature field as well [31]. Here we have to distinguish between the integration of the assignment flow [81] and integration schemes for numerically

solving (1.4). The task of designing the latter schemes faces the *differentiate-then-discretize* vs. *discretize-then-differentiate* dilemma. Conditions and ways to resolve this dilemma have been studied in the optimal control literature [28, 61]. See also the recent survey [62] and references therein.

From a more distant viewpoint, our work ties in with research on networks from a *dynamical systems* point of view, that emanated from [34] in computer science and has also recently been promoted in mathematics [24]. The recent work [27], for example, studied stability issues of discrete-time network dynamics using techniques of numerical ODE integration. The authors adopted the discretize-then-differentiate viewpoint on the parameter estimation problem and suggested symplectic numerical integration in order to achieve better stability. As mentioned above, our work contrasts in that inference is always *exact* during learning, unlike the more involved architecture of [27] where learning is based on *approximate* inference. Furthermore, in our case, symplectic numerical integration is a *consequence* of making the diagram of Figure 2.2 (page 30) *commute*. This property qualifies our approach as a proper (though rudimentary) method of *optimal control* (cf. [61]).

1.3 Outline and Contribution

The main contributions of this thesis are:

- **A novel approach to MAP inference for discrete graphical models.**
  This approach fits well into the smooth geometric framework [8] and its key ingredient is a *smooth* approximation

$$E_\tau(\mu_\mathcal{V}) = \langle \theta_\mathcal{V}, \mu_\mathcal{V} \rangle + \sum_{ij \in \mathcal{E}} d_{\theta_{ij},\tau}(\mu_i, \mu_j), \qquad \tau > 0 \tag{1.5}$$

  of the LP-relaxation (1.3), where $d_{\theta_{ij},\tau}$ denotes the *smoothed Wasserstein distance* between the discrete label assignment measures μ_i, μ_j coupled along the edge ij of the underlying graph. These Wasserstein distances properly take into account the pairwise regularization parameters θ_{ij} of the labeling problem (1.2). Our approach restricts the function E_τ to the so-called assignment manifold and iteratively determines a labeling by tightly combining geometric optimization with rounding to an integral solution in a smooth fashion. This formulation constitutes an inner relaxation of the discrete labeling problem, i.e. throughout this process local marginalization constraints are satisfied. This novel inference approach is worked out in detail in Chapter 4.

- **A novel approach to model parameter learning for the assignment flow.** We tackle the parameter learning problem for image labeling by focusing on the smooth geometric approach [8] and ignore the connection to discrete graphical models. This problem was raised in [8, Section 5 and Fig. 14], and we present a detailed solution. The key idea is to formulate the learning problem for image labeling as a specific instance of the general parameter estimation problem (1.4). Thereby, the parameters p determine the *linear assignment flow* (1.4b) [81] whose solution is evaluated by a suitable loss function (1.4a) at some point of time T. Since the formulation (1.4) is well-known in the communities of scientific computing and optimal control, we can draw on a rich literature to study and solve this problem. The output of optimizing (1.4) are optimal parameters which determine the regularization property of the assignment flow. By construction, the objective (1.4a) is based on given ground truth images. In order to tackle *novel* images (no ground truth labeling is given) we propose a simple predictor map that is based on the solution of (1.4). This function takes *unseen* data and predicts the regularization parameters for the linear assignment flow, i.e. no optimization is involved. All details of this novel approach are worked out in detail in Chapter 5.

This thesis is organized as follows: After this outline follows an introduction of the basic notation (Section 1.4).

In **Chapter 2** we collect the relevant mathematical background material for this thesis. We start by recalling the basic definitions, ideas and theorems from differential geometry that serve as reference for Chapter 3 (Section 2.1). Afterwards, we continue with the introduction of probabilistic graphical models (Section 2.2). In particular we highlight the details of the linear programming (LP) relaxation (Section 2.2.4) and the corresponding loopy belief propagation inference algorithm (Section 2.2.5). We end this chapter by providing the necessary background on parameter estimation of dynamical systems (Section 2.3). Hereby, we focus on sensitivity analysis from a continuous perspective (Section 2.3.1) as well as on the practical perspective (Section 2.3.3), i.e. the numerical approximation.

Chapter 3 summarizes the basic ideas of the smooth geometric approach of [8] for image labeling. Since this chapter provides the basis for the two subsequent chapters (4 & 5) containing our main contributions, we collect the material in detail. In particular we recall the definition of the so-called assignment manifold (Section 3.1), and present the general framework of [65] for numerically integrating vector fields on the assignment manifold (Section 3.2). Image labeling is performed by following

the so-called assignment flow. We present the main components of this flow (Section 3.3), followed by numerical experiments (Section 3.4). We end this chapter by a brief overview of several extensions (Section 3.5).

In **Chapter 4** we present how a given discrete graphical model can be evaluated in the smooth geometric setting explained in Chapter 3. We start by reformulating the LP-relaxation (1.3) onto the assignment manifold and smoothly approximate the resulting functional (Section 4.1). The explicit expressions of the corresponding gradient are provided afterwards (Sections 4.2 & 4.3). We continue with the details of combining relaxation and rounding into a *single* process (Section 4.4.1) and discuss the connections to established belief propagation (Section 4.4.2). We end this chapter by discussing the results of four different types of experiments (Section 4.5).

Chapter 5 is devoted to our second main contribution, namely a novel approach to model parameter learning for the assignment flow. We start by proposing our problem formulation (Section 5.1), calculating the expressions of the corresponding gradients and differentials (Section 5.2) and continuing with our optimization strategy (Section 5.3). Finally, we discuss the results of two experiments that highlight the model expressiveness of the assignment flow as well as limitations that result from learning *constant* parameters (Section 5.4).

1.4 Notation

In this section we introduce the basic notation, sorted by subject area. A list of symbols can be found in the nomenclature on p. 153.

Indexing and Operations. All vectors are regarded as column vectors, and $x^\top$ denotes transposition of a vector x. We ignore transposition however when vectors are explicitly specified by their components; e.g. we write $x = (y, z)$ instead of the more cumbersome $x = (y^\top, z^\top)^\top$. We set $[n] = \{1, 2, \dots, n\}$ for $n \in \mathbb{N}$. Given a matrix

$$A = \begin{pmatrix} A_1 \\ \vdots \\ A_m \end{pmatrix} = \left(A^1 \dots A^n\right) \in \mathbb{R}^{m \times n}, \tag{1.6}$$

we denote the row vectors by A_i, $i \in [m]$ and the column vectors by A^j, $j \in [n]$, whereas superscripts in brackets, e.g. $A_i^{(k)}$, index iterative steps.

The functions exp and log apply *component-wise* to strictly positive vectors $x \in \mathbb{R}^n_{++}$, e.g. $e^x = (e^{x_1}, \dots, e^{x_n})$, and similarly for strictly positive matrices. Likewise, if

$x, y \in \mathbb{R}^n_{++}$, we simply write

$$x \cdot y = (x_1 y_1, \ldots, x_n y_n), \qquad \frac{x}{y} = \Big(\frac{x_1}{y_1}, \ldots, \frac{x_n}{y_n}\Big), \tag{1.7}$$

for the *component-wise* multiplication and division.

If A is a $m \times n$ matrix and B is a $p \times q$ matrix, then the *Kronecker product* $A \otimes B$ is a $mp \times nq$ block matrix of the form:

$$A \otimes B = \begin{bmatrix} a_{11}B & \cdots & a_{1n}B \\ \vdots & \ddots & \vdots \\ a_{m1}B & \cdots & a_{mn}B \end{bmatrix}. \tag{1.8}$$

The canonical *matrix inner product* is $\langle A, B \rangle = \operatorname{tr}(A^\top B)$, where tr denotes the *trace of a matrix*, i.e.

$$\operatorname{tr}(A^\top B) = \sum_{i \in [m]} \langle A_i, B_i \rangle = \sum_{j \in [n]} \langle A^j, B^j \rangle = \sum_{i \in [m], j \in [n]} A_{ij} B_{ij}. \tag{1.9}$$

The inner product $\langle x, y \rangle = \sum_{i \in [n]} x_i y_i$ denotes the *Euclidean inner product*.

Graphs. An *undirected graph* $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ consists of a set of *vertices* $\mathcal{V}$ and a set of *edges* $\mathcal{E}$. The adjacency relation $i \sim j$ means that vertices i and j are connected by an undirected edge $ij \in \mathcal{E}$, where the latter denotes the *unordered* pair $\{i, j\} = ij = ji$. The graph $\mathcal{G}$ is turned into a *directed graph* by assigning an *orientation* to every edge ij, which then form *ordered* pairs $(i, j) \neq (j, i)$. By abuse of notation we sometimes write $(i, j) = ij$ in the oriented case, however, the exact meaning will be clear from context. We only consider graphs *without multiple* edges between any pair of nodes $i, j \in \mathcal{V}$. The *neighborhood* of vertex i is given by the set

$$\mathcal{N}(i) = \{j \in \mathcal{V} \colon i \sim j\} \tag{1.10}$$

of all vertices adjacent to i, and its cardinality $d(i) = |\mathcal{N}(i)|$ is the *degree* of i.

Differentials and Jacobian. Let $(\mathcal{M}, g)$ be a Riemannian manifold with metric g, and a smooth function $f \colon \mathcal{M} \to \mathbb{R}$, the *Riemannian gradient* of f is denoted by $\operatorname{grad} f$. The differential of f is denoted by df. More generally, for a map $F \colon \mathcal{M} \to \mathcal{N}$ between manifolds, we write $dF_p[v] \in T_{F(p)}\mathcal{N}$, $p \in \mathcal{M}$, $v \in T_p\mathcal{M}$, if the base point p matters.

In the *Euclidean* case $f \colon \mathbb{R}^n \to \mathbb{R}$, the gradient is a column vector and denoted by ∇f. For $F \colon \mathbb{R}^n \to \mathbb{R}^m$, we identify the differential $dF \in \mathbb{R}^{m \times n}$ with the Jacobian matrix.

If $x = (x_1, x_2)^\top \in \mathbb{R}^n = \mathbb{R}^{n_1} \times \mathbb{R}^{n_2}$ with $n = n_1 + n_2$, then the Jacobian of $F(x) = F(x_1, x_2)$ with respect to the parameter vector x_i is denoted by $d_{x_i} F$, for $i = 1, 2$.

Coupling measure. We set $\mathbb{1}_n = (1, 1, \ldots, 1)^\top \in \mathbb{R}^n$. The *probability simplex* $\Delta_n = \{p \in \mathbb{R}^n_+ : \langle \mathbb{1}_n, p \rangle = 1\}$ contains all discrete distributions on $[n]$. A doubly stochastic matrix $\mu_{ij} \in \mathbb{R}^{n \times n}_+$, also called *coupling measure* in this thesis in connection with discrete optimal transport, has the property: $\mu_{ij}\mathbb{1}_n \in \Delta_n$ and $\mu_{ij}^\top \mathbb{1}_n \in \Delta_n$. We denote the two *marginal distributions* of μ_{ij} by μ_i and μ_j, respectively, and the linear mapping for extracting them by

$$\mathcal{A} \colon \mathbb{R}^{n \times n} \to \mathbb{R}^{2n}, \qquad \mu_{ij} \mapsto \mathcal{A}\mu_{ij} = \begin{pmatrix} \mu_{ij}\mathbb{1}_n \\ \mu_{ij}^\top \mathbb{1}_n \end{pmatrix} = \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix}. \tag{1.11a}$$

Its transpose is given by

$$\mathcal{A}^\top \colon \mathbb{R}^{2n} \to \mathbb{R}^{n \times n}, \qquad (\nu_i, \nu_j) \mapsto \mathcal{A}^\top \begin{pmatrix} \nu_i \\ \nu_j \end{pmatrix} = \nu_i \mathbb{1}_n^\top + \mathbb{1}_n \nu_j^\top. \tag{1.11b}$$

The *kernel (nullspace)* of the linear mapping $\mathcal{A}$ is denoted by $\ker(\mathcal{A})$ and its *image* by $\operatorname{im}(\mathcal{A})$.

Convex Analysis. A subset $C \subset \mathbb{R}^n$ is *convex* if

$$(1 - \lambda)x + \lambda y \in C, \quad \forall x, y \in C, \quad \forall \lambda \in [0, 1]. \tag{1.12}$$

The *affine hull* of some set $C \subset \mathbb{R}^n$ is the set of all affine combinations of its points,

$$\operatorname{aff} C = \{\lambda_1 x_1 + \cdots + \lambda_d x_d : \lambda_1 + \cdots + \lambda_d = 1, x_1, \ldots, x_d \in C\}. \tag{1.13}$$

The *relative interior* of a non-empty convex set C is defined by

$$\operatorname{relint}(C) = \{x \in \operatorname{aff} C : \exists \varepsilon > 0 \text{ such that } (x + \varepsilon \mathbb{B}(0)) \cap \operatorname{aff} C \subset C\}. \tag{1.14}$$

Various orthogonal projections onto a convex set are generally denoted by Π and distinguished by a corresponding subscript, like $\Pi_C, \Pi_{\mathcal{P}}, \cdots$, etc.

The *log-exponential* function $\operatorname{logexp}_\varepsilon \in \mathcal{F}_0$ is defined as

$$\operatorname{logexp}_\varepsilon(x) := \varepsilon \log\Big(\sum_{i \in [n]} e^{\frac{x_i}{\varepsilon}} \Big). \tag{1.15a}$$

It uniformly approximates the function vecmax $\in \mathcal{F}_0$ [59, Ex. 1.30], i.e.

$$\lim_{\varepsilon \searrow 0} \operatorname{logexp}_\varepsilon(x) = \operatorname{vecmax}(x) = \max\{x_i\}_{i\in[n]}. \tag{1.15b}$$

We use the following basic result from convex analysis, where $\partial f(x)$ denotes the subdifferential of a function $f \in \mathcal{F}_0$ at x.

Theorem 1.1 (inversion rule for subgradients; [59, Prop. 11.3])
Let $f \in \mathcal{F}_0$. Then

$$\hat{p} \in \partial f(\hat{x}) \quad \Leftrightarrow \quad \hat{x} \in \partial f^*(\hat{p}) \quad \Leftrightarrow \quad f(\hat{x}) + f^*(\hat{p}) = \langle \hat{p}, \hat{x} \rangle. \tag{1.16}$$

△

In Section 4.3.1 we apply the following classical theorem of Danskin and its extension by Rockafellar.

Theorem 1.2 ([22, 58])
Let $f(z) = \max_{w\in W} g(z,w)$, where W is compact and the function $g(\cdot, w)$ is differentiable and $\nabla_z g(z,w)$ is continuously depending on (z,w). If in addition $g(z,w)$ is convex in z, and if $\overline{z}$ is a point such that $\arg\max_{w\in W} g(\overline{z},w) = \{\overline{w}\}$, then f is differentiable at $\overline{z}$ with

$$\nabla f(\overline{z}) = \nabla_z g(\overline{z}, \overline{w}). \tag{1.17}$$

△

Chapter 2

Mathematical Background

In this chapter, we collect background material that is used throughout this thesis. First, we summarize the basic definitions and concepts of differential geometry (Section 2.1). We continue with the introduction of probabilistic graphical models (Section 2.2). Especially, we focus on discrete graphical models and the corresponding inference and learning problem. This section ends with the derivation of the LP relaxation and a brief sketch of loopy belief propagation that is an approximated algorithm for the inference task. The last topic of this chapter is the study of parameter estimation of dynamical systems (Section 2.3). We explain the problem of analyzing the sensitivity of dynamical systems with respect to parameter, as well as symplectic partitioned Runge–Kutta methods that are well-suited numerical integration techniques for this problem.

2.1 Elements of Differential Geometry

We briefly summarize the relevant material on *differential geometry*. We refer the reader for a broad introduction to this field to the standard works [45, 46, 47] which served as reference for this section. For a more algorithm focused line we refer the reader to [2].

Let $\mathcal{M}$ be a topological space. If $\mathcal{M}$ is paracompact, Hausdorff, second countable and locally homeomorphic to $\mathbb{R}^d$, then we call $\mathcal{M}$ a *topological d-manifold*.

A *chart* on $\mathcal{M}$ is a pair $(\mathcal{U},\phi)$, where $\mathcal{U} \subset \mathcal{M}$ is a subset and $\phi\colon \mathcal{U} \to \mathbb{R}^d$ is a homeomorphism. The component functions $(x^1,\dots,x^d)$ of map ϕ, with $\phi(p) = (x^1(p),\dots,x^d(p))$, are called *local coordinates* of $p \in \mathcal{U}$.

A *smooth atlas* A is a collection of charts $(\mathcal{U}_i,\phi_i)$ whose domains cover $\mathcal{M}$, i.e. $\bigcup_i \mathcal{U}_i = \mathcal{M}$. In addition, the charts have to be *smoothly compatible*, that is for two charts $(\mathcal{U}_i,\phi_i)$ and $(\mathcal{U}_j,\phi_j)$ with $\mathcal{U}_i \cap \mathcal{U}_j \neq \emptyset$ the *transition map* $\phi_j \circ \phi_i^{-1}$ is a diffeomorphism. A smooth atlas A on $\mathcal{M}$ is *maximal* if any chart which is smoothly compatible with every chart in A is already in A. We call a maximal smooth atlas A a *smooth structure on* $\mathcal{M}$.

A *smooth manifold* consists of a pair $(\mathcal{M}, A)$, where $\mathcal{M}$ is a topological manifold and A is a smooth structure on $\mathcal{M}$.

For any point $p \in \mathcal{M}$, the *tangent space* to $\mathcal{M}$ at point p is a real vector space and denoted by $T_p\mathcal{M} \subset \mathbb{R}^n$. The *tangent bundle* $T\mathcal{M}$ of a smooth manifold $\mathcal{M}$ is defined as

$$T\mathcal{M} = \coprod_{p \in \mathcal{M}} T_p\mathcal{M}, \tag{2.1}$$

which is the disjoint union of the tangent spaces at all points of $\mathcal{M}$.

Let $\mathcal{M}$ and $\mathcal{N}$ be smooth manifolds and $\phi\colon \mathcal{M} \to \mathcal{N}$ a smooth map. Then, the *pushforward (differential)* of ϕ at $p \in \mathcal{M}$ is a linear map

$$d\phi_p\colon T_p\mathcal{M} \to T_{\phi(p)}\mathcal{N}, \tag{2.2}$$

and is defined by using a velocity vector

$$d\phi_p[v] = \frac{d}{dt}\phi(\gamma(t))\bigg|_{t=0}, \tag{2.3}$$

where $\gamma\colon (-\epsilon, \epsilon) \to \mathcal{M}$ is a smooth curve with $\gamma(0) = p, \dot{\gamma}(0) = v \in T_p\mathcal{M}$.

Suppose $\mathcal{M}$ and $\mathcal{N}$ are smooth manifolds and $\phi\colon \mathcal{M} \to \mathcal{N}$ is a differentiable map. If ϕ is a bijection and its inverse $\phi^{-1}\colon \mathcal{N} \to \mathcal{M}$ is differentiable, then ϕ is called a *diffeomorphism*. If there exists a diffeomorphism ϕ between smooth manifolds $\mathcal{M}$ and $\mathcal{N}$, then we say that $\mathcal{M}$ and $\mathcal{N}$ are *diffeomorphic* or simply write $\mathcal{M} \cong \mathcal{N}$.

2.1.1 Vector Fields and Integral Curves

A *vector field* on M is a smooth map

$$X\colon \mathcal{M} \to T\mathcal{M}, \quad p \mapsto X_p, \tag{2.4}$$

with the property that $X_p \in T_p\mathcal{M}$. The collection of all smooth vector fields on $\mathcal{M}$ is denoted by $\mathfrak{X}(\mathcal{M})$.

Suppose $\phi\colon \mathcal{M} \to \mathcal{N}$ is a smooth map, X a vector field on $\mathcal{M}$ and Y a vector field on $\mathcal{N}$ such that for each point $p \in \mathcal{M}$:

$$d\phi_p[X_p] = Y_{\phi(p)}. \tag{2.5}$$

Then, we say that the vector fields X and Y are *ϕ-related*.

Proposition 2.1 ([45, Proposition 8.19])
Suppose $\phi\colon \mathcal{M} \to \mathcal{N}$ is a diffeomorphism between smooth manifolds $\mathcal{M}$ and $\mathcal{N}$, and X be a vector field on $\mathcal{M}$. Then, there exists a unique smooth vector field on $\mathcal{N}$ which is ϕ-related to X. This unique vector field is called *pushforward of X by ϕ* and is given by

$$(\phi_* X)_q := d\phi_{\phi^{-1}(q)}\left[X_{\phi^{-1}(q)}\right]. \tag{2.6}$$

This formula originates from the following diagram

$$\begin{array}{ccc} \mathcal{M} & \xrightarrow{X} & T\mathcal{M} \\ \downarrow{\scriptstyle \phi} & & \downarrow{\scriptstyle d\phi} \\ \mathcal{N} & \xrightarrow{\phi_* X} & T\mathcal{N} \end{array} \qquad \triangle$$

If $X \in \mathfrak{X}(\mathcal{M})$, an *integral curve of X* is a differentiable curve $\gamma\colon J \to \mathcal{M}$ whose velocity at each point is equal to the value of X at that point:

$$\gamma'(t) = X_{\gamma(t)}, \quad \forall\, t \in J. \tag{2.7}$$

For $0 \in J$, the point $\gamma(0)$ is called *initial point* of γ. We also use the term *trajectory* of X for the curve $\gamma(t)$.

Proposition 2.2 (Naturality of Integral Curves; [45, Proposition 9.6])
Suppose $\mathcal{M}$ and $\mathcal{N}$ are smooth manifolds and $\phi\colon \mathcal{M} \to \mathcal{N}$ is a smooth map. Then $X \in \mathfrak{X}(\mathcal{M})$ and $Y \in \mathfrak{X}(\mathcal{N})$ are ϕ-related if and only if ϕ takes integral curves of X to integral curves of Y, i.e. for each integral curve ψ of X, $\phi \circ \psi$ is an integral curve of Y. $\triangle$

2.1.2 Connections

Connections are rules for taking derivatives of vector fields in a coordinate-independent fashion.

Let X, Y be smooth vector fields on a smooth manifold $\mathcal{M}$, and let $f\colon \mathcal{M} \to \mathbb{R}$ be a smooth function. Then, the *Lie bracket* is defined by

$$[X, Y]f = XYf - YXf, \tag{2.8}$$

and the operator $[X, Y]$ is again a vector field.

An *affine connection* on $\mathcal{M}$ is a map

$$\nabla\colon \mathfrak{X}(\mathcal{M}) \times \mathfrak{X}(\mathcal{M}) \to \mathfrak{X}(\mathcal{M}), \qquad (X, Y) \mapsto \nabla_X Y, \tag{2.9}$$

satisfying

(i) $\nabla_X Y$ is linear over $C^\infty(\mathcal{M})$ in X : for $f_1, f_2 \in C^\infty(\mathcal{M})$ and $X_1, X_2 \in \mathfrak{X}(\mathcal{M})$,

$$\nabla_{f_1 X_1 + f_2 X_2} Y = f_1 \nabla_{X_1} Y + f_2 \nabla_{X_2} Y. \tag{2.10}$$

(ii) $\nabla_X Y$ is linear over $\mathbb{R}$ in Y : for $a_1, a_2 \in \mathbb{R}$ and $Y_1, Y_2 \in \mathfrak{X}(\mathcal{M})$,

$$\nabla_X (a_1 Y_1 + a_2 Y_2) = a_1 \nabla_X Y_1 + a_2 \nabla_X Y_2. \tag{2.11}$$

(iii) ∇ satisfies the following product rule: for $f \in C^\infty(M)$,

$$\nabla_X (f Y) = f \nabla_X Y + (X f) Y. \tag{2.12}$$

The expression $\nabla_X Y$ denotes the *covariant derivative of* Y *in the direction* X.

If an affine connection ∇ satisfies the following product rule

$$\nabla_X (g(Y, Z)) = g(\nabla_X Y, Z) + g(Y, \nabla_X Z), \tag{2.13}$$

for all vector fields X, Y, Z, we say that it is *compatible with* g.

If the torsion of an affine connection ∇ vanishes, i.e.

$$\nabla_X Y - \nabla_Y X \equiv [X, Y], \tag{2.14}$$

where $[X, Y]$ denotes the *Lie bracket*, we say that ∇ is *symmetric*.

Suppose $\mathcal{M}$ is a smooth manifold, and $\gamma \colon I \to \mathcal{M}$ is a smooth curve. Then, a *vector field along curve* γ is a continuous map

$$V \colon I \to T\mathcal{M} \qquad \text{such that} \quad V(t) \in T_{\gamma(t)}\mathcal{M},\ \forall t \in I. \tag{2.15}$$

A connection ∇ induces a unique operator which takes the *covariant derivative along curve* γ (see [46, Theorem 4.24] for more details).

Geodesics and Exponential Maps

If a smooth curve γ has zero acceleration, i.e.

$$\nabla_{\dot\gamma} \dot\gamma \equiv 0 \tag{2.16}$$

then γ is called a *geodesic*. In order to see the dependency of γ with respect to the chosen metric g, we rewrite (2.16) in terms of *local coordinates* (x^i). Then, it follows

that $\gamma(t) = \left(x^1(t), \ldots, x^n(t)\right)$ is a *geodesic* if and only if its components satisfy

$$\ddot{x}^k(t) + \dot{x}^i(t)\dot{x}^j(t)\Gamma^k_{ij}(x(t)) = 0. \tag{2.17}$$

This system of second-order ODEs is called *geodesic equation*. The coefficients Γ^k_{ij} are commonly called *Christoffel symbols* or *connection coefficients*. These coefficients depend on the metric g and are given by

$$\Gamma^k_{ij} = \frac{1}{2} g^{kl}\left(\partial_i g_{jl} + \partial_j g_{il} - \partial_l g_{ij}\right), \tag{2.18}$$

where g_{ij} are called *metric coefficients* and g^{kl} (with superscripts) denotes the corresponding inverse matrix.

The *exponential map* is defined by

$$\mathrm{Exp}_p \colon V_p \to \mathcal{M}, \quad v \mapsto \mathrm{Exp}_p(v) = \gamma_v(1), \quad p \in \mathcal{M}. \tag{2.19a}$$

with domain

$$V_p = \left\{v \in T_p\mathcal{M} : \gamma_v(t) \in \mathcal{M}, t \in [0,1]\right\}, \tag{2.19b}$$

and geodesic γ with $\gamma_v(0) = p$ and $\dot{\gamma}_v(0) = v$.

2.1.3 Riemannian Geometry

A *Riemannian manifold* $(\mathcal{M}, g^{\mathcal{M}})$ is a smooth manifold $\mathcal{M}$ whose tangent spaces are endowed with a smoothly varying inner product $g^{\mathcal{M}}$ which is called *Riemannian metric*. If the Riemannian manifold is clear from the context, we simplify notation and write $g_p(u,v) = \langle u, v\rangle_p$. We will use interchangeably the notation $g^{\mathcal{M}}_p(u,v) = \langle u, v\rangle^{\mathcal{M}}_p$.

Suppose $\phi \colon \mathcal{M} \to \mathcal{N}$ is a diffeomorphism between smooth manifolds $\mathcal{M}$ and $\mathcal{N}$, where ($\mathcal{N}$ is equipped with a Riemannian metric $g^{\mathcal{N}}$). Then, we can define a Riemannian metric on $\mathcal{M}$ by *pulling back* the metric $g^{\mathcal{N}}$. This is why the metric

$$\left(\phi^* g^{\mathcal{N}}_p\right)(\cdot,\cdot) \colon T_p\mathcal{M} \times T_p\mathcal{M} \to \mathbb{R}, \quad (u,v) \mapsto \left(\phi^* g^{\mathcal{N}}_p\right)(u,v) := g^{\mathcal{N}}_{\phi(p)}\left(d\phi(u), d\phi(v)\right), \tag{2.20}$$

is called the *pullback metric* for all $u, v \in T_p\mathcal{M}$.

An *isometry* is a diffeomorphism $\phi \colon (\mathcal{M}, g^{\mathcal{M}}) \to (\mathcal{N}, g^{\mathcal{N}})$ between Riemannian manifolds which pulls back the metric, i.e. $\phi^* g^{\mathcal{N}} = g^{\mathcal{M}}$.

Theorem 2.3 (Fundamental Lemma of Riemannian Geometry; [47, Theorem 5.4.]) Let $(\mathcal{M}, g)$ be a Riemannian manifold. There exists a unique affine connection ∇ on $\mathcal{M}$ that is compatible with g and symmetric. This connection is called *Riemannian connection* or the *Levi–Civita connection* of g. △

If $f\colon \mathcal{M} \to \mathbb{R}$ is a smooth function on a smooth Riemannian manifold $(\mathcal{M}, g)$, then the *Riemannian gradient* $\operatorname{grad}_{\mathcal{M}} f \in \mathfrak{X}(\mathcal{M})$ is the vector field defined by

$$\langle \operatorname{grad}_{\mathcal{M}} f, v\rangle_p = df_p[v], \quad \forall v \in T_p\mathcal{M}, \quad p \in \mathcal{M}. \tag{2.21}$$

Geodesics and exponentials maps, which are constructed by using the Levi–Civita connection, are called *Riemannian geodesic* and *Riemannian exponential map*, respectively.

2.2 Probabilistic Graphical Models

A *probabilistic graphical model* uses a graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ to model the conditional dependencies between random variables. As this representation is very flexible, these models are used in many different disciplines. A treatment of this topic from the variational viewpoint can be found in [74].

2.2.1 Discrete Graphical Models

An important classical family of probabilistic models is the *exponential model*

$$p(x|\theta) = \exp\big(\langle\theta, \phi(x)\rangle - \psi_p(\theta)\big) = \frac{1}{Z(\theta)} \exp\big(\langle\theta, \phi(x)\rangle\big), \tag{2.22}$$

where θ are called the *canonical parameters* and $\phi(x)$ are known as the *potential functions* or *sufficient statistics*. The normalizing constant $Z(\theta)$ is called *partition function* and is given by

$$Z(\theta) = \int_{\mathcal{X}} \exp\langle\theta, \phi(x)\rangle. \tag{2.23}$$

The function $\psi_p(\theta)$ is the *log-partition function*. As the name suggests, it is given by

$$\psi_p(\theta) = \log Z(\theta). \tag{2.24}$$

We obtain a related *energy* of the probability (2.22) by

$$E(x) = \log p(x|\theta) = \langle\theta, \phi(x)\rangle - \psi_p(\theta). \tag{2.25}$$

Next, we introduce an important member of the class of models (2.22), namely the *discrete graphical model*. Let the variables $x = (x_i)_{i\in[m]}$, with $m := |\mathcal{V}|$, be indexed by the vertex set $\mathcal{V}$. The edge set $\mathcal{E}$ models the conditional dependencies between these variables, i.e. if the edge $ij \in \mathcal{E}$ exists, the variables x_i and x_j explicitly dependent on each other. In discrete graphical models, the variables take values from the *discrete set*

$$x_i \in \mathcal{X} = \{\ell_1, \dots, \ell_n\}, \tag{2.26}$$

which we call *labels* or *prototypes*. A global assignment of these variables is denoted by

$$x \in \mathcal{X}^m = \prod_{i\in[m]} \mathcal{X}_i, \tag{2.27}$$

with $\mathcal{X}_i$ given by (2.26). In this setting, we associate with the joint probability $p(x|\theta)$ the *energy function*

$$p(x|\theta) = \frac{1}{Z(\theta)} \exp(-E(x)), \qquad E(x) = \sum_{i\in\mathcal{V}} \theta_i(x_i) + \sum_{ij\in\mathcal{E}} \theta_{ij}(x_i, x_j), \tag{2.28}$$

where the potentials θ are scalar functions consisting of *unary potentials*

$$\theta_i \colon \mathcal{X} \to \mathbb{R}, \qquad x_i \mapsto \theta_i(x_i), \tag{2.29a}$$

and *pairwise potentials*

$$\theta_{ij} \colon \mathcal{X} \times \mathcal{X} \to \mathbb{R}, \qquad (x_i, x_j) \mapsto \theta_{ij}(x_i, x_j). \tag{2.29b}$$

The potentials assign a scalar cost to the chosen label $x_i \in \mathcal{X}$ at node $i \in \mathcal{V}$, and to the edge $ij \in \mathcal{E}$ with chosen label $x_i, x_j \in \mathcal{X}$, respectively. We call the values of θ the *model parameters* of $p(x|\theta)$.

2.2.2 Inference

Suppose the model parameters θ of a discrete graphical model $p(x|\theta)$ are given. Then, the term *inference* corresponds to the following two problems:

1. *Marginalization:* We wish to compute the marginal distribution for some subset $\mathcal{A} \subset \mathcal{V}$, i.e. we are interested in

$$p(x_A|\theta) = \sum_{x \in \mathcal{X}^{\mathcal{V}\setminus\mathcal{A}}} p(x|\theta), \tag{2.30}$$

   where A defines a subset of variables x_A.

2. *Maximum a posteriori (MAP) inference:* In order to find the most probable configuration of $p(x|\theta)$ we consider the problem

$$x^* = \operatorname*{argmax}_{x \in \mathcal{X}^m} p(x|\theta). \tag{2.31a}$$

   Maximizing the probability $p(x|\theta)$ corresponds to minimizing the associated *discrete energy*

$$x^* = \operatorname*{argmin}_{x \in \mathcal{X}^m} E(x), \quad E(x) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{ij \in \mathcal{E}} \theta_{ij}(x_i, x_j). \tag{2.31b}$$

   Since for given parameters θ the log-partition function $\psi_p(\theta)$ is constant, we dropped the term in (2.31b).

The difficulty of these problems highly depends on the underlying graph structure $\mathcal{G} = (\mathcal{V}, \mathcal{E})$. In general, the pairwise interactions of the discrete energy function makes (2.31b) a combinatorially hard task.

Remark 2.1 For discrete graphical models the integral in the partition function (2.23) becomes a sum. Unfortunately, the combinatorially large number of summands makes the exact evaluation of the partition function $Z(\theta)$ and log-partition function $\psi(\theta)$ intractable. If the underlying graph is acyclic, the exact evaluation of $Z(\theta)$ and $\psi(\theta)$ is tractable. △

2.2.3 Learning

The counter problem to inference is *learning*. The learning task deals with the setting that the *empirical mean parameters*

$$\hat{\mu} := \frac{1}{n} \sum_{i=1}^{n} \phi(x_i) \tag{2.32}$$

are given, and we wish to learn the model parameters θ. In other words, $\hat{\mu}$ corresponds to given example data and we wish to learn the parameters of our model in such a way that the observed data (2.32) is well represented by our model.

More specifically, the corresponding *log-likelihood* of (2.22) reads

$$\frac{1}{n}\log\prod_{i=1}^{n} p(x_i|\theta) = \frac{1}{n}\sum_{i=1}^{n}\left(\langle\theta,\phi(x_i)\rangle - \psi_p(\theta)\right) \overset{(2.32)}{=} \langle\theta,\hat{\mu}\rangle - \psi_p(\theta). \tag{2.33}$$

Then, the *model parameter learning* problem consists of maximizing (2.33) with respect to the *model parameters* θ, i.e.

$$\theta^* = \operatorname*{argmax}_{\theta\in\Theta}\left\{\langle\theta,\hat{\mu}\rangle - \psi_p(\theta)\right\}, \tag{2.34}$$

where Θ denotes the *parameter space*.

Remark 2.2 (Inference effects learning) The learning problem (2.34) is related to the MAP inference problem (2.31a), but with a crucial difference: the log-partition function $\psi_p(\theta)$ has to be evaluated. In view of Remark 2.1, this makes learning a difficult problem for general graphs. In addition to this, a subroutine of solving the learning problem is inference. If the underlying inference problem can not be solved exactly, but only approximatively, the learning process is definitely effected. △

2.2.4 Linear Programming Relaxation

In this thesis we focus on the MAP inference problem (2.31b). Since this problem is a combinatorially hard problem, a major class of algorithms is based on the *linear programming (LP) relaxation* [78]. This relaxation technique consists of two steps: The first step is to reformulate (2.31b) as an integer linear program, and the second step is to replace the integer constraints with linear ones. Thus, the NP-hard problem (2.31b) is transformed into a solvable *linear program*.

To make these steps in detail, we introduce additional notation needed in subsequent sections. First, we encode the values of the discrete objective function (2.31b) by defining *local model parameter vectors* and *matrices*

$$\theta_i := \big(\theta_i(\ell_k)\big)_{k\in[n]} \in \mathbb{R}^n, \qquad \theta_{ij} := \big(\theta_{ij}(\ell_k,\ell_r)\big)_{k,r\in[n]} \in \mathbb{R}^{n\times n}, \tag{2.35}$$

with $\ell_k,\ell_r \in \mathcal{X}$, and where the indices are given by the vertices $i \in \mathcal{V}$ and edges $ij \in \mathcal{E}$ of the underlying graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$. Likewise, we assemble all these local terms into

the vectors

$$\theta := (\theta_{\mathcal{V}}, \theta_{\mathcal{E}}), \qquad \text{where} \qquad \theta_{\mathcal{V}} := (\theta_i)_{i\in\mathcal{V}}, \qquad \text{and} \qquad \theta_{\mathcal{E}} := (\theta_{ij})_{ij\in\mathcal{E}}, \tag{2.36}$$

where we interchangeably regard $\theta_{ij} \in \mathbb{R}^{n^2}$ either as local vector or as local matrix $\theta_{ij} \in \mathbb{R}^{n\times n}$, depending on the context. Next we define *local indicator vectors*

$$\mu_i := \big(\mu_i(\ell_k)\big)_{k\in[n]} \in \{0,1\}^n, \quad \mu_{ij} := \big(\mu_{ij}(\ell_k,\ell_r)\big)_{k,r\in[n]} \in \{0,1\}^{n\times n}, \tag{2.37}$$

with $\ell_k, \ell_r \in \mathcal{X}$, and indexed in the same way as (2.35) and assembled into the vectors

$$\mu := (\mu_{\mathcal{V}}, \mu_{\mathcal{E}}), \qquad \text{where} \qquad \mu_{\mathcal{V}} := (\mu_i)_{i\in\mathcal{V}}, \qquad \text{and} \qquad \mu_{\mathcal{E}} := (\mu_{ij})_{ij\in\mathcal{E}}. \tag{2.38}$$

To ensure that μ consistently represents valid labelings, the variables have to satisfy the so-called *marginalization constraints*

$$\sum_{x_i\in\mathcal{X}} \mu_i(x_i) = 1, \qquad \forall i \in \mathcal{V}, \tag{2.39a}$$

$$\sum_{x_j\in\mathcal{X}} \mu_{ij}(x_i,x_j) = \mu_i, \qquad \forall ij \in \mathcal{E}, \forall x_i \in \mathcal{X}, \tag{2.39b}$$

$$\sum_{x_i\in\mathcal{X}} \mu_{ij}(x_i,x_j) = \mu_j, \qquad \forall ij \in \mathcal{E}, \forall x_j \in \mathcal{X}. \tag{2.39c}$$

These constraints define with the integer constraints (2.37) the *marginal polytope*

$$\mathcal{M}_{\mathcal{G}} := \operatorname{conv}\big\{\mu\colon \mu \text{ satisfies (2.37) and (2.39)}\big\}. \tag{2.40}$$

The combinatorial optimization problem (2.31b) is now in the form of an *integer linear program*: $\min_{\mu\in\mathcal{M}_{\mathcal{G}}} \langle\theta,\mu\rangle$.

As mentioned above, the LP relaxation consists of replacing the integrality constraints (2.37) by the convex polyhedral sets

$$\mu_i \in \Delta_n, \quad \mu_{ij} \in \Pi(\mu_i,\mu_j), \qquad i \in \mathcal{V},\ ij \in \mathcal{E}, \tag{2.41a}$$

$$\Pi(\mu_i,\mu_j) := \big\{\mu_{ij} \in \mathbb{R}_+^{n\times n}\colon \mu_{ij}\mathbb{1} = \mu_i,\ \mu_{ij}^\top\mathbb{1} = \mu_j,\ \mu_i,\mu_j \in \Delta_n\big\}. \tag{2.41b}$$

The resulting relaxation of problem (2.31b) reads

$$\min_{\mu\in\mathcal{L}_{\mathcal{G}}} \langle\theta,\mu\rangle = \min_{\mu\in\mathcal{L}_{\mathcal{G}}} \langle\theta_{\mathcal{V}},\mu_{\mathcal{V}}\rangle + \langle\theta_{\mathcal{E}},\mu_{\mathcal{E}}\rangle, \tag{2.42}$$

where the so-called *local polytope* $\mathcal{L}_\mathcal{G}$ is the set of all vectors μ of the form (2.38) with components ranging over the sets specified by (2.41). The term "local" refers to the local marginalization constraints (2.41b).

Remark 2.3 The marginal polytope $\mathcal{M}_\mathcal{G} \subseteq \mathcal{L}_\mathcal{G}$ is a subset of the local polytope. It can be shown that if the underlying graph $\mathcal{G}$ is acyclic, the local polytope $\mathcal{L}_\mathcal{G}$ is identical to the marginal polytope $\mathcal{M}_\mathcal{G}$. We refer to [50] for background and details. △

2.2.5 Loopy Belief Propagation

In this section we briefly sketch *belief propagation (BP)* and the origin of corresponding *messages*. For a detailed derivation of BP we refer the reader to Appendix B and for background and more details we refer to [80, 74]. A study of *inference techniques* for solving the discrete minimization problem (2.31b) can be found in [39].

Starting point is the primal linear program (LP) (2.42) written in the form

$$\min_{\mu\in\mathcal{L}_\mathcal{G}} \langle\theta,\mu\rangle = \min_{\mu} \langle\theta,\mu\rangle \quad \text{subject to} \quad A\mu = b,\ \mu \geq 0, \tag{2.43}$$

where the constraints represent the feasible set $\mathcal{L}_\mathcal{G}$ which is explicitly given by the local marginalization constraints (2.39). The corresponding dual LP reads

$$\max_{\nu} \langle b,\nu\rangle, \quad A^\top \nu \leq \theta, \tag{2.44}$$

with dual (multiplier) variables

$$\nu = (\nu_\mathcal{V}, \nu_\mathcal{E}) = (\ldots, \nu_i, \ldots, \nu_{ij}(x_i), \ldots, \nu_{ij}(x_j), \ldots), \quad i \in \mathcal{V}, \quad ij \in \mathcal{E} \tag{2.45}$$

corresponding to the affine primal constraints. In order to obtain a condition that relates optimal vectors μ and ν without subdifferentials that are caused by the non-smoothness of these LPs, we consider the *smoothed* primal problem

$$\min_{\mu\in\mathcal{L}_\mathcal{G}} \langle\theta,\mu\rangle - \varepsilon H(\mu), \quad \varepsilon > 0, \qquad H(\mu) = \sum_{ij\in\mathcal{E}} H(\mu_{ij}) - \sum_{i\in\mathcal{V}} \big(d(i)-1\big) H(\mu_i) \tag{2.46}$$

with smoothing parameter $\varepsilon > 0$. The function $H(\mu)$ denotes the *Bethe entropy* with degree $d(i) = |\mathcal{N}(i)|$ of vertex i and with local entropy functions

$$H(\mu_i) = -\sum_{x_i\in\mathcal{X}} \mu_i(x_i) \log \mu_i(x_i), \qquad H(\mu_{ij}) = -\sum_{x_i,x_j\in\mathcal{X}} \mu_{ij}(x_i,x_j) \log \mu_{ij}(x_i,x_j). \tag{2.47}$$

Setting temporarily $\varepsilon = 1$, and evaluating the optimality condition $\nabla_\mu L(\mu, \nu) = 0$ based on the corresponding Lagrangian

$$L(\mu, \nu) = \langle \theta, \mu \rangle - H(\mu) + \langle \nu, A\mu - b \rangle \tag{2.48}$$

yields the relations connecting μ and ν,

$$\mu_i(x_i) = e^{\nu_i} e^{-\theta_i(x_i)} \prod_{j \in \mathcal{N}(i)} e^{\nu_{ij}(x_i)}, \qquad x_i \in \mathcal{X},\ i \in \mathcal{V}, \tag{2.49a}$$

$$\mu_{ij}(x_i, x_j) = e^{\nu_i + \nu_j} e^{-\theta_{ij}(x_i,x_j) - \theta_i(x_i) - \theta_j(x_j)} \prod_{k \in \mathcal{N}(i)\setminus\{j\}} e^{\nu_{ik}(x_i)} \prod_{k \in \mathcal{N}(j)\setminus\{i\}} e^{\nu_{jk}(x_j)}, \tag{2.49b}$$

for $x_i, x_j \in \mathcal{X},\ ij \in \mathcal{E}$. The terms $e^{\nu_i}, e^{\nu_i + \nu_j}$ normalize the expressions on the right-hand side whereas the so-called *messages* $e^{\nu_{ij}(x_i)}$ enforce the local marginalization constraints $\mu_{ij} \in \Pi(\mu_i, \mu_j)$. Invoking these latter constraints enables us to eliminate the left-hand side of (2.49) to obtain after some algebra (cf. Appendix B) the fixed point equations

$$e^{\nu_{ij}(x_i)} = e^{\nu_j} \sum_{x_j \in \mathcal{X}} \Big(e^{-\theta_{ij}(x_i,x_j) - \theta_j(x_j)} \prod_{k \in \mathcal{N}(j)\setminus\{i\}} e^{\nu_{jk}(x_j)} \Big), \qquad ij \in \mathcal{E}, \quad x_i \in \mathcal{X}, \tag{2.50}$$

solely in terms of the *dual* variables, commonly called *sum-product algorithm* or *loopy belief propagation* by *message passing*. Repeating this derivation, after weighting the entropy function $H(\mu)$ of (2.48) by ε as in (2.46), and taking the limit $\lim_{\varepsilon \searrow 0}$, yields relation (2.50) with the sum replaced by the max operation. This is a consequence of taking the log of both sides and relation (1.15b) of the log-exponential function. This fixed point iteration is called *max-product algorithm* in the literature.

2.3 Parameter Estimation of Dynamical Systems

Throughout this section we provide the necessary background on the optimization of the following *parameter estimation problem*

$$\min_{p \in \mathcal{P}} \quad \mathcal{C}\big(x(T)\big) \tag{2.51a}$$

$$\text{s.t.} \quad \dot{x}(t) = f(x(t), p, t), \quad t \in [0, T], \tag{2.51b}$$

$$x(0) = x_0, \tag{2.51c}$$

where $\mathcal{C}\colon \mathbb{R}^{n_x} \to \mathbb{R}$ is a smooth objective function. The constraints are given by a general *initial value problem (IVP)*, which consist of a system of ordinary differential equations (ODEs) (2.51b) that is parametrized by a vector $p \in \mathcal{P} \subset \mathbb{R}^{n_p}$, and an initial value $x_0 \in \mathbb{R}^{n_x}$ (2.51c). To ensure existence, uniqueness and continuous differentiability of the solution trajectory $x(t)$ on the whole time horizon $[0, T]$, we assume that $f(\cdot, p, \cdot)$ of (2.51c) is Lipschitz continuous on $\mathbb{R}^{n_x} \times [0, T]$, for any p.

By assuming the initial value x_0 and the time horizon $[0, T]$ to be fixed, the objective function (2.51a) effectively is a function of the parameter p, i.e.

$$\Phi\colon \mathbb{R}^{n_p} \to \mathbb{R}, \quad \Phi(p) := \mathcal{C}(x(T, p)). \tag{2.52}$$

In order to minimize (2.52) with a gradient based method, we have to compute the *gradient*

$$\nabla_p \Phi(p) = d_p x(T, p)^\top \nabla_x \mathcal{C}(x(T, p)). \tag{2.53}$$

The term $d_p x(T, p)$ is called *sensitivity*, hence it measures the sensitivity of the the solution trajectory $x(t)$ at time T with respect to changes in the parameter p. Two basic approaches for determining (2.53) are stated in Section 2.3.1, and we briefly highlight why using one of them, the *adjoint approach*, is advantageous for computing sensitivities. In Section 2.3.2, we recall symplectic Runge-Kutta methods and conditions for preserving quadratic invariants. The latter property relates to the derivation of a class of numerical methods such that evaluating (2.53), which derives from the time-continuous problem (2.51), is *identical* to first discretizing (2.51) followed by computing the corresponding derived expression (2.53). In Section 2.3.4 we derive two specific instances of the general numerical scheme in detail.

2.3.1 Sensitivity Analysis

This section shows how the sensitivity $d_p x(T, p)$ in (2.53) can be determined by solving one of the two initial value problems defined below: the *variational system* and the *adjoint system*.

Theorem 2.4 (Variational System; [32, Chapter I.14, Theorem 14.1])
Suppose the derivatives $d_x f$ and $d_p f$ exist and are continuous in the neighborhood of the solution $x(t)$ for $t \in [0, T]$. Then the sensitivity with respect to the parameters

$$d_p x(T, p) =: \delta(T) \tag{2.54}$$

exists, is continuous and satisfies the *variational system*

$$\dot{\delta}(t) = d_x f(x(t), p, t)\delta(t) + d_p f(x(t), p, t), \quad t \in [0, T], \tag{2.55a}$$
$$\delta(0) = 0 \in \mathbb{R}^{n_x \times n_p}, \tag{2.55b}$$

with $\delta(t) \in \mathbb{R}^{n_x \times n_p}$. If the initial value $x(0)$ of (2.51c) depends on the parameters p, the initial value (2.55b) has to be adjusted as $\delta(0) = d_p x(0)$.

Proof See Appendix A.1. □

For the computation of the variational system (2.55) the solution $x(t)$ is required. The variational system (2.55) is a matrix-valued system of dimension $n_x \times n_p$, and therefore the size of the system grows with the number of parameters n_p. Nevertheless, for small n_p the variational system can simultaneously be integrated numerically with system (2.51b).

Theorem 2.5 (Adjoint System)
Suppose the derivatives $d_x f$ and $d_p f$ exist and are continuous in the neighborhood of the solution $x(t)$ for $t \in [0, T]$. Then the sensitivity with respect to the parameters is given by

$$d_p x(T, p)^\top = \int_0^T d_p f(x(t), p, t)^\top \lambda(t) dt, \tag{2.56}$$

where $\lambda(t) \in \mathbb{R}^{n_x \times n_x}$ solves the adjoint system

$$\dot{\lambda}(t) = -d_x f(x(t), p, t)^\top \lambda(t), \quad t \in [0, T], \tag{2.57a}$$
$$\lambda(T) = I \in \mathbb{R}^{n_x \times n_x}. \tag{2.57b}$$

Proof This proof is elaborated on in a broader context in Section 2.3.3. □

Similar to the variational system of Theorem 2.4, solving the adjoint system (2.57) requires the solution $x(t)$. The adjoint system is matrix-valued of dimension $n_x \times n_x$, in contrast to the variational system which is of dimension $n_x \times n_p$. Thus, if $n_p \gg n_x$, as will be the case in our scenario, it is more efficient to solve (2.57) instead of (2.55). Another major difference is that the adjoint system is defined *backwards* in time, starting from the endpoint T. This has important computational advantages for our setting. In view of the required gradient (2.53), we are not interested in the full sensitivity but rather in the derivative along the direction $\eta := \nabla_x \mathcal{C}(x(T, p))$, i.e. $d_p x(T, p)^\top \eta$. This can be achieved by exploiting the structure of the adjoint system,

by multiplying (2.57) from the right by η and setting $\overline{\lambda}(t) := \lambda(t)\eta$. The resulting IVP is again an adjoint system, no longer being matrix-valued but vector-valued $\overline{\lambda}(t) \in \mathbb{R}^{n_x}$ with $\overline{\lambda}(T) = \eta \in \mathbb{R}^{n_x}$. Therefore, we consider the latter case and denote $\overline{\lambda}(t)$ again by $\lambda(t)$, which is vector-valued.

Remark 2.4 As a consequence, we will focus on the adjoint system (2.57) in the remainder of this section. In particular, (2.56) will be used to estimate parameters p by solving (2.51) using a gradient descent flow. This requires to solve the adjoint system numerically. However, a viable alternative to this *differentiate-then-discretize* approach is to reverse this order, that is to *discretize* problem (2.51) first, and then to derive a corresponding *time-discrete* adjoint system (*differentiate*). It turns out that both ways are equivalent if a proper class of numerical integration scheme is chosen for discretizing the system in time. This will be shown in Section 2.3.3 after collecting required background material in Section 2.3.2. △

2.3.2 Symplectic Partitioned Runge–Kutta Methods

In this section we recall basic concepts of numerical integration from [31, 62] in order to prepare Section 2.3.3. Symplectic schemes are typically applied to Hamiltonian systems in order to conserve certain quantities, often with a physical background. The pseudo-Hamiltonian defined below by (2.68) will play a similar role, albeit there is no physical background for our concrete scenario to be studied in subsequent sections.

A general *s-stage Runge–Kutta (RK) method* with $s \in \mathbb{N}$ is given by [30, Ch. II]

$$x_{n+1} = x_n + h_n \sum_{i=1}^{s} b_i k_{n,i}, \quad h_n = t_{n+1} - t_n, \tag{2.58a}$$

$$k_{n,i} = f(X_{n,i}, p, t_n + c_i h_n), \tag{2.58b}$$

$$X_{n,i} = x_n + h_n \sum_{j=1}^{s} a_{ij} k_{n,j}. \tag{2.58c}$$

The coefficients $a_{ij}, b_i, c_i \in \mathbb{R}$ can be arranged in a so-called Butcher tableau (Fig. 2.1), with entries a_{ij} defining the Runge–Kutta matrix A.
Suppose the Runge–Kutta matrix A is lower-triangular (see Fig. 2.1, RIGHT), i.e.

$$a_{ij} = 0 \quad \text{for} \quad j \geq i, \tag{2.59}$$

then the resulting RK schemes is called *explicit*, since (2.58b) can be evaluated explicitly. In contrast, if A is not lower triangular, (2.58b) can not be solved explicitly

$$\begin{array}{c|cccc} c_1 & a_{11} & a_{12} & \dots & a_{1s} \\ c_2 & a_{21} & a_{22} & \dots & a_{2s} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss} \\ \hline & b_1 & b_2 & \dots & b_s \end{array} = \begin{array}{c|c} c & A \\ \hline & b^T \end{array}, \qquad \begin{array}{c|ccccc} c_1 & & & & & \\ c_2 & a_{21} & & & & \\ c_3 & a_{31} & a_{32} & & & \\ \vdots & \vdots & \vdots & \ddots & & \\ c_s & a_{s1} & a_{s2} & \dots & a_{ss-1} & \\ \hline & b_1 & b_2 & \dots & b_{s-1} & b_s \end{array}$$

Figure 2.1: Butcher tableau of a Runge–Kutta method. LEFT: Tableau of a general s-stage RK method. RIGHT: Tableau of an explicit s-stage RK method.

as a system of algebraic equations has to be solved. Therefore, these methods are called *implicit* RK methods. They are well-suited for the numerical integration of stiff ODEs, but are also significantly more complex than explicit ones. For more details on these methods we refer the reader to [30, Ch. II.7] and for a thorough treatment of *stiff problems* we refer to [29, Ch. IV]. The following theorem specifies a condition to the step-size h under which a solution of the equations (2.58b) exists.

Theorem 2.6 (Existence of a Numerical Solution; [30, Ch. II, Theorem 7.2])
For any $p \in \mathbb{R}^{n_p}$ let $f(\cdot, p, \cdot)$ of (2.51c) be continuous and satisfy a Lipschitz condition on $\mathbb{R}^{n_x} \times [0, T]$ with constant L, independent of p. If

$$h < \frac{1}{L \max_{i=1,\dots,s} \sum_{j=1}^{s} |a_{ij}|}, \tag{2.60}$$

then there exists a unique solution of (2.58), which can be obtained by iteration. If $f(x, p, t)$ is q times differentiable, the functions k_i (as functions of h) are also in C^q.

Proof A detailed proof can be found in [30, Chapter II, Theorem 7.2]. □

Suppose a given system (2.51b) is *partitioned* into two parts with $x = (q^\top, p^\top)^\top$, $f = (f_1^\top, f_2^\top)^\top$ and

$$\dot{q} = f_1(q, p, t), \tag{2.61a}$$
$$\dot{p} = f_2(q, p, t). \tag{2.61b}$$

Then we can integrate this system by using two different sets of coefficients

$$a_{ij}, b_i, c_i \in \mathbb{R} \text{ for (2.61a)}, \qquad \overline{a}_{ij}, \overline{b}_i, \overline{c}_i \in \mathbb{R} \text{ for (2.61b)}. \tag{2.62}$$

These methods are commonly called *partitioned Runge–Kutta methods*. The theorems 2.7 and 2.8 state conditions under which those methods preserve certain quantities that should be invariant under the flow of the given system. In this sense such RK schemes are called *symplectic*.

Theorem 2.7 (Symplectic Runge–Kutta Method; [31, Ch. VI, Theorems 7.6 & 7.10]) Assume that system (2.51b) has a quadratic invariant I, i.e. $I(\cdot,\cdot)$ is a real-valued bilinear mapping such that $(d/dt)I(x(t),x(t)) = 0$, for each t and x_0. If the coefficients of a Runge–Kutta method (2.58) satisfy

$$b_i a_{ij} + b_j a_{ji} - b_i b_j = 0, \tag{2.63}$$

then the value $I(x_n, x_n)$ does not depend on n. △

Theorem 2.8 (Symplectic Partitioned RK Method; [62, Theorems 2.4 and 2.6]) Assume that $S(\cdot,\cdot)$ is a real-valued bilinear mapping such that $(d/dt)S(q(t),p(t)) = 0$ for each t and x_0 of the solution $x(t) = [q(t)^\top, p(t)^\top]^\top$ of (2.61). If the coefficients of the partitioned Runge–Kutta method (2.62) satisfy

$$b_i\overline{a}_{ij} - b_i\overline{b}_j + \overline{b}_j a_{ji} = 0, \quad \overline{b}_i = b_i, \quad \overline{c}_i = c_i, \tag{2.64}$$

then the value $S(q_n, p_n)$ does not depend on n. △

Remark 2.5 Assume the first set of Runge–Kutta coefficients are given by a_{ij}, b_i, c_i with indices $i, j \in [s]$ and are used for the first n-variables (2.61a). Furthermore, let $b_i \neq 0$ for all stages $i \in [s]$. Then in view of condition (2.64) we can construct a *symplectic* PRK method by choosing

$$\overline{a}_{ij} := b_j - b_j a_{ji}/b_i, \quad \overline{b}_i := b_i, \quad \overline{c}_i := c_i, \tag{2.65}$$

as coefficients for the second n-variables (2.61b). This construction results in an overall *symplectic PRK method* of the partitioned system (2.61). △

2.3.3 Computing Adjoint Sensitivities

In this section we come back to Remark 2.4 about two basic approaches for computing the adjoint sensitivity (2.53), namely the *differentiate-then-discretize* approach and the *discretize-then-differentiate* approach. Figure 2.2 illustrates both approaches by paths colored with *blue* and *violet*, respectively. We work out the details of each path in this section. Thereby, our main objective is to make this diagram *commutative* by adopting a class of numerical schemes as outlined in the preceding section.

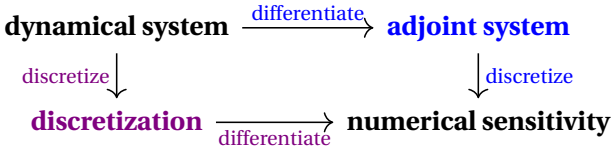


Figure 2.2: Computing adjoint sensitivities. By adopting a class of Runge–Kutta methods, we show that the diagram *commutes*, i.e. *identical* results are obtained either if the continuous problem is differentiated first and than discretized *(blue path)*, or the other way around *(violet path)*.

In order to simplify notation, we drop the dependency of $x(t)$ on the parameter p and just write $x(t)$. The following theorem details the *blue path* of Figure 2.2.

Theorem 2.9 (Adjoint Sensitivity: Differentiate-then-Discretize)
The gradient (2.53) of objective function (2.52) with respect to the parameter p is given by

$$\nabla\Phi(p) = \int_0^T d_p f(x(t), p, t)^\top \lambda(t)\, dt, \tag{2.66}$$

where $x(t), \lambda(t)$ solve the *two-point boundary value problem (BVP)*

$$\dot{x}(t) = f(x(t), p, t), \qquad x(0) = x_0, \tag{2.67a}$$
$$\dot{\lambda}(t) = -d_x f(x(t), p, t)^\top \lambda(t), \qquad \lambda(T) = \nabla\mathcal{C}(x(T)). \tag{2.67b}$$

In terms of the pseudo-Hamiltonian

$$H(x, \lambda, p, t) = \langle f(x, p, t), \lambda \rangle, \tag{2.68}$$

the system has the following form

$$\dot{x}(t) = \quad d_\lambda H(x, \lambda, p, t), \qquad x(0) = x_0, \tag{2.69a}$$
$$\dot{\lambda}(t) = -d_x H(x, \lambda, p, t), \qquad \lambda(T) = \nabla\mathcal{C}(x(T)). \tag{2.69b}$$

Proof See Appendix A.1. □

Remark 2.6 The presence of the pseudo-Hamiltonian (2.68) suggests to use either a *symplectic* RK method or a *symplectic* PRK method to integrate the BVP (2.67). In view of Remark 2.5, we can use a general RK method with coefficients a_{ij}, b_i, c_i for $i, j \in [s]$ for the first variables (2.67a), and another RK method with $\overline{a}_{ij}, \overline{b}_i, \overline{c}_i$ for $i, j \in [s]$ satisfying (2.65) for the second variables (2.67b). Again, this construction results in an overall *symplectic PRK method* of the BVP (2.67). △

Now we consider the alternative *violet path* of Figure 2.2. The application of a RK method with step-sizes $h_n = t_{n+1} - t_n > 0$ to problem (2.51) results in the *nonlinear optimization problem*

$$\min_{p \in \mathcal{P}} \quad \mathcal{C}\big(x_N(p)\big) \tag{2.70a}$$

$$\text{s.t.} \quad x_{n+1} = x_n + h_n \sum_{i=1}^{s} b_i k_{n,i}, \quad n = 0, \dots, N-1, \quad x_0 = x(0), \tag{2.70b}$$

$$k_{n,i} = f(X_{n,i}, p, t_n + c_i h_n), \quad i \in [s], \tag{2.70c}$$

$$X_{n,i} = x_n + h_n \sum_{j=1}^{s} a_{ij} k_{n,j}, \quad i \in [s]. \tag{2.70d}$$

Next, we *differentiate* this problem and state the result in the following theorem.

Theorem 2.10 (Adjoint Sensitivity: Discretize-then-Differentiate Approach) Suppose the step-size h_n satisfies condition (2.60). Then, the gradient of the objective function $\Phi(p) = \mathcal{C}(x_N(p))$ from (2.70) with respect to parameter p is given by

$$\nabla \Phi(p) = \sum_{n=0}^{N-1} h_n \sum_{i=1}^{s} \overline{b}_i \left(d_p f(X_{n,i}, p, t_n + \overline{c}_i h_n) \right)^\top \Lambda_{n,i}, \tag{2.71}$$

where the discrete adjoint variables are given by

$$\lambda_{n+1} = \lambda_n + h_n \sum_{i=1}^{s} \overline{b}_i \ell_{n,i}, \quad h_n = t_{n+1} - t_n, \quad n = 0, \dots, N-1, \tag{2.72a}$$

$$\ell_{n,i} = -d_x f(X_{n,i}, p, t_n + \overline{c}_i h_n)^\top \Lambda_{n,i}, \quad i \in [s] \tag{2.72b}$$

$$\Lambda_{n,i} = \lambda_n + h_n \sum_{j=1}^{s} \overline{a}_{ij} \ell_{n,j}, \quad i \in [s], \tag{2.72c}$$

with initial value $\lambda_N = \nabla \mathcal{C}(x_N)$ and the internal stages $X_{n,i}$ are given by (2.70d). This scheme is a general Runge–Kutta method (2.58a)-(2.58c) applied to the adjoint sys-

tem (2.67b) with $b_i \neq 0$, $i = 1, \dots, s$, and coefficients

$$\overline{a}_{ij} = b_j - \frac{a_{ji} b_j}{b_i}, \quad \overline{b}_i = b_i, \quad \overline{c}_i = c_i, \qquad \text{for } i, j = 1, \dots, s. \tag{2.73}$$

Proof An outline of the proof can be found in [62, Theorem 3.6]. Following the suggested outline, we provide a detailed proof in Appendix A.1. □

The proof of this theorem is based on the following lemma, which is a slightly different version of Lemma 3.5 in [62]. The strategy is to state the Lagrangian of the nonlinear problem (2.70) and derive all formulas of Theorem 2.10 by a straightforward application of the lemma. Again, for a detailed proof we refer the reader to Appendix A.1.

Lemma 2.11 (cf. [62, Lemma 3.5])
Suppose the mapping $\phi\colon \mathbb{R}^{n_p \times d'} \to \mathbb{R}^{d'}$ is such that the Jacobian matrix $d_\gamma \phi$ is invertible at a point $(p_0, \gamma_0) \in \mathbb{R}^{n_p} \times \mathbb{R}^{d'}$, that is in the neighborhood of p_0, the equation $\phi(p, \gamma) = 0$ defines γ as a function of p. For some given function $\mathcal{C}\colon \mathbb{R}^{n_p \times d'} \to \mathbb{R}$ consider the induced function of the form $\Phi\colon \mathbb{R}^{n_p} \to \mathbb{R}$, defined by $\Phi(p) := \mathcal{C}(p, \gamma(p))$. We introduce the Lagrangian

$$\mathcal{L}(p, \gamma, \lambda) = \mathcal{C}(p, \gamma) + \langle \phi(p, \gamma), \lambda \rangle. \tag{2.74}$$

Then the Euclidean gradient of Φ with respect to p at p_0 is given by

$$\nabla \Phi(p_0) = \nabla_p \mathcal{L}(p_0, \gamma_0, \lambda_0), \tag{2.75}$$

where the vectors $\gamma_0 = \gamma(p_0) \in \mathbb{R}^{d'}$ and $\lambda_0 \in \mathbb{R}^{d'}$ are uniquely determined by

$$0 = \nabla_\lambda \mathcal{L}(p_0, \gamma_0, \lambda_0) = \phi(p_0, \gamma_0), \tag{2.76a}$$

$$0 = \nabla_\gamma \mathcal{L}(p_0, \gamma_0, \lambda_0) \quad \Longleftrightarrow \quad \nabla_\gamma \mathcal{C}(p_0, \gamma_0) = -d_\gamma \phi(p_0, \gamma_0)^\top \lambda_0. \tag{2.76b}$$

Proof See Appendix A.1. □

Remark 2.7 Comparing the statements of Theorem 2.9 and Theorem 2.10, we see that the formula of the discrete sensitivity (2.71) is an approximation of the integral (2.66) with quadrature weights b_i. Furthermore, we observe that the coefficients of

the constructed PRK method (2.65) coincides with the derived coefficients (2.73). Thus, by restricting the class of numerical schemes to *symplectic PRK methods* satisfying (2.64), the approaches due to the Theorem 2.9 (and Remark 2.6) and Theorem 2.10 are mathematically identical, and the diagram depicted by Figure 2.2 *commutes*. △

2.3.4 Adjoint Sensitivity: Two Specific Numerical Schemes

In this section we complement and illustrate the general results of the preceding section by specifying two numerical schemes in detail.

Explicit Euler method

First, we consider the *explicit Euler method* [32] for the forward integration of the dynamics (2.67a). The straightforward use of (2.65) leads to another set of Runge–Kutta coefficients for integrating the adjoint system (2.67b). In combination these *forward* and *backward* coefficients form an overall symplectic partitioned Runge–Kutta method and are given by Table 2.1.

$$\begin{array}{c|c} c_1 & a_{11} \\ \hline & b_1 \end{array} = \begin{array}{c|c} 0 & \\ \hline & 1 \end{array} \qquad \begin{array}{c|c} \overline{c}_1 & \overline{a}_{11} \\ \hline & \overline{b}_1 \end{array} = \begin{array}{c|c} 0 & 1 \\ \hline & 1 \end{array}$$

forward coefficients *backward coefficients*

Table 2.1: Symplectic PRK coefficients induced by the explicit Euler method.

We derive concrete formulas for the integration of the adjoint system by substituting the backward coefficients $\overline{a}_{11}, \overline{b}_1$ and $\overline{c}_1$ from Table 2.1 into (2.72), which gives

$$\lambda_{n+1} = \lambda_n + h_n \ell_{n,1} \tag{2.77a}$$
$$\ell_{n,1} = -\partial_x f(X_{n,1}, t_n)^\top \Lambda_{n,1} \tag{2.77b}$$
$$\Lambda_{n,1} = \lambda_n + h_n \ell_{n,1}. \tag{2.77c}$$

Since (2.77c) coincides with (2.77a), we can plug (2.77b) into (2.77a) and rewrite the overall schemes (2.77) by traversing from $n+1$ to n. Then the resulting scheme is given by

$$\lambda_n = \lambda_{n+1} + h_n d_x f(X_{n,1}, t_n)^\top \lambda_{n+1}, \tag{2.78}$$

which is an *explicit method traversing backwards from* $n+1$ *to* n. Since $\overline{b}_1 = 1$ and $s = 1$ (number of stages), the formula of the discrete gradient (2.71) reads

$$\nabla\Phi(p) = \sum_{n=0}^{N-1} h_n d_p f(X_{n,1}, t_n)^\top \lambda_{n+1}. \tag{2.79}$$

Heun's method

Now, we integrate the dynamics (2.67a) with *Heun's method* [32]. Again, the straightforward use of (2.65) leads to another Runge–Kutta method for integrating the adjoint system (2.67b). Both *forward* and *backward* coefficients of the overall symplectic partitioned Runge–Kutta method are given by Table 2.2. Although the butcher tableau of the backward coefficients (see Table 2.2, right matrix) is completely dense, the final update formulas are *explicit*, as we show below.

c_1	a_{11}	a_{12}	0		
c_2	a_{21}	a_{22}	1	1	
	b_1	b_2		1/2	1/2

forward coefficients

$\overline{c}_1$	$\overline{a}_{11}$	$\overline{a}_{12}$	0	1/2	-1/2
$\overline{c}_2$	$\overline{a}_{21}$	$\overline{a}_{22}$	1	1/2	1/2
	$\overline{b}_1$	$\overline{b}_2$		1/2	1/2

backward coefficients

Table 2.2: Symplectic PRK coefficients induced by Heun's method.

Again, we derive the concrete formulas of the PRK method by substituting the backward coefficients from Table 2.2 into (2.72)

$$\lambda_{n+1} = \lambda_n + h_n\left(\tfrac{1}{2}\ell_{n,1} + \tfrac{1}{2}\ell_{n,2}\right) \tag{2.80a}$$
$$\ell_{n,1} = -d_x f(X_{n,1}, t_n)^\top \Lambda_{n,1} \tag{2.80b}$$
$$\ell_{n,2} = -d_x f(X_{n,2}, t_n + h_n)^\top \Lambda_{n,2} \tag{2.80c}$$
$$\Lambda_{n,1} = \lambda_n + h_n\left(\tfrac{1}{2}\ell_{n,1} - \tfrac{1}{2}\ell_{n,2}\right) \tag{2.80d}$$
$$\Lambda_{n,2} = \lambda_n + h_n\left(\tfrac{1}{2}\ell_{n,1} + \tfrac{1}{2}\ell_{n,2}\right). \tag{2.80e}$$

Note, that (2.80e) coincides with (2.80a), which implies the equations

$$\lambda_{n+1} = \Lambda_{n,2} \quad \text{and} \quad \ell_{n,2} = -d_x f(X_{n,2}, t_n + h_n)^\top \lambda_{n+1}. \tag{2.81}$$

Using (2.81), we reformulate (2.80d)

$$\Lambda_{n,1} \;=\; \lambda_n + h_n\left(\tfrac{1}{2}\ell_{n,1} - \tfrac{1}{2}\ell_{n,2}\right) = \lambda_n + h_n\left(\tfrac{1}{2}\ell_{n,1} - \tfrac{1}{2}\ell_{n,2}\right) + (h_n\ell_{n,2} - h_n\ell_{n,2})$$

$$= \lambda_n + h_n\left(\tfrac{1}{2}\ell_{n,1} + \tfrac{1}{2}\ell_{n,2}\right) - h_n\ell_{n,2} \overset{(2.80a)}{=} \lambda_{n+1} - h_n\ell_{n,2}$$
$$\overset{(2.81)}{=} \lambda_{n+1} + h_n d_x f(X_{n,2}, t_n + h_n)^\top \lambda_{n+1}. \tag{2.82a}$$

Formula (2.82a) is an *explicit Euler step traversing backwards from $n+1$ to n*. Thus, we can rewrite the overall scheme (2.80) as

$$\tilde{\lambda}_n = \lambda_{n+1} + h_n d_x f(X_{n,2}, t_n + h_n)^\top \lambda_{n+1} \tag{2.83a}$$
$$\lambda_n = \lambda_{n+1} + \frac{h_n}{2}\left(d_x f(X_{n,1}, t_n)^\top \tilde{\lambda}_n + d_x f(X_{n,2}, t_n + h_n)^\top \lambda_{n+1}\right), \tag{2.83b}$$

which is again an *explicit method traversing backwards from $n+1$ to n*. By plugging the coefficients $\overline{b}_1 = \overline{b}_2 = 1$ and $s = 2$ (number of stages) into (2.71) we obtain the formula of the discrete gradient

$$\nabla_p \Phi(p) = \sum_{n=0}^{N-1} \frac{h_n}{2}\left(d_p f(X_{n,1}, t_n)^\top \tilde{\lambda}_n + d_p f(X_{n,2}, t_n + h_n)^\top \lambda_{n+1}\right). \tag{2.84}$$

Chapter 3

Image Labeling by Assignment

In this chapter we summarize the work of [8] which tackles the image labeling problem by a *smooth geometric* approach. Since this approach forms the basis of Chapter 4 & 5, we present the ideas and concepts in detail. We start by introducing the main mathematical object, the so-called *assignment manifold* (Section 3.1), and explain afterwards how a given flow on this manifold can be transformed onto the tangent space (Section 3.2). The image labeling task is performed by following the so-called *assignment flow*, which is a smooth flow evolving on the assignment manifold. We present the main components of this flow (Section 3.3), followed by numerical experiments (Section 3.4). We end this chapter by a brief overview of several extensions (Section 3.5). For summarizing recent work based on the assignment flow and a discussion of further aspects we refer to [69] .

The work [8] was primarily introduced to perform image labeling, but it is not limited to that task. More precisely, all objects and components are defined *locally*, i.e. in terms of the local graph structure $\mathcal{G}$. Therefore, the approach is directly applicable to arbitrary graph structures $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ without further modifications.

3.1 The Assignment Manifold

In this section we introduce the *main* mathematical object of this thesis: the so-called *assignment manifold* that consists of smaller building blocks, namely the relative interior of the probability simplex. We summarize without proofs the relevant material on the geometry of the probability simplex (Section 3.1.1) and continue afterwards with the definition of the assignment manifold (Section 3.1.2). All proofs of Section 3.1.1 can be found in the respective appendix of [8].

3.1.1 Local Object: Relative Interior of the Probability Simplex

The image labeling task consists of assigning to each node $i \in \mathcal{V}$ one of n predefined labels $\mathcal{X}$. By taking a probabilistic view of this assignment we model the decision for

each node $i \in \mathcal{V}$ as a point on the relative interior of the probability simplex

$$\mathcal{S}_n := \operatorname{relint}(\Delta_n) = \{p \in \Delta_n : \, p_i > 0 \text{ for } i = 1, \ldots, n\} \tag{3.1}$$

with constant tangent space

$$T_p\mathcal{S}_n = \{v \in \mathbb{R}^n : \, \langle v, \mathbb{1}_n \rangle = 0\} =: T_n \subset \mathbb{R}^n, \quad p \in \mathcal{S}_n. \tag{3.2}$$

Since $\langle v, \mathbb{1}_n \rangle = 0$ for all $v \in T_n$, there is an orthogonal decomposition $\mathbb{R}^n = T_n \oplus \mathbb{R}\mathbb{1}_n$. The *orthogonal projection* onto T_n is given by

$$\Pi_{T_n} : \mathbb{R}^n \to T_n, \quad x \mapsto \Pi_{T_n}(x) = x - \frac{1}{n}\langle \mathbb{1}_n, x \rangle \mathbb{1}_n = \left(I - \frac{1}{n}\mathbb{1}_n \mathbb{1}_n^\top \right) x. \tag{3.3}$$

In this setting each vertex $j \in [n]$ of $\mathcal{S}_n$ corresponds to the unique decision of one specific label $x_j \in \mathcal{X}$, $j \in [n]$. The *barycenter* of $\mathcal{S}_n$ is given by the uniform distribution $\frac{1}{n}\mathbb{1}_n \in \mathcal{S}_n$ and is denoted by $\mathbb{1}_{\mathcal{S}_n} := \frac{1}{n}\mathbb{1}_n$.

By endowing the probability space $\mathcal{S}_n$ at each $p \in \mathcal{S}_n$ with the Fisher–Rao metric

$$g_p^{\mathcal{S}_n} : \, T_n \times T_n \to \mathbb{R}, \quad (u, v) \mapsto g_p^{\mathcal{S}_n}(u, v) := \langle \frac{u}{\sqrt{p}}, \frac{v}{\sqrt{p}} \rangle, \quad \forall\, u, v \in T_n, \tag{3.4}$$

it becomes a differentiable Riemannian manifold $(\mathcal{S}_n, g^{\mathcal{S}_n})$.

In order to get a sense for the induced geometry of $\mathcal{S}_n$, we consider the scaled sphere $\mathcal{N} = 2\mathbb{S}^{n-1}$ which can be regarded as a manifold with the Riemannian metric induced by the Euclidean inner product of $\mathbb{R}^n$. The manifolds $\mathcal{S}_n$ and $\mathcal{N}$ are diffeomorphic with the following map

$$\psi : \mathcal{S}_n \to \mathcal{N}, \quad p \mapsto s = \psi(p) := 2\sqrt{p}. \tag{3.5}$$

We call the diffeomorphism (3.5) *sphere-map* ψ. This map is illustrated in Fig. 3.1 for the 2-dimensional simplex $\mathcal{S}_2$.

For $(\mathcal{S}_n, g^{\mathcal{S}_n})$ and $(\mathcal{N}, g^{\mathcal{N}})$, where $g^{\mathcal{N}}$ denotes the Riemannian metric induced by the standard Euclidean inner product of $\mathbb{R}^n$, it can be shown that the sphere-map ψ (3.5) is also an *isometry* (see [8, Appendix 2]). That is, the map ψ preserves lengths of tangent vectors and curves. Furthermore, by recalling the definition of an *isometry*, we can understand the Fisher–Rao metric (3.4) as the induced *pullback metric* of the Euclidean inner product, i.e. $\psi^* g^{\mathcal{N}} = g^{\mathcal{S}_n}$.

The *Riemannian gradient* of a smooth function $f : \mathcal{S}_n \to \mathbb{R}$ is defined as follows. Let $f : \mathcal{S}_n \to \mathbb{R}$ be a smooth function on $\mathcal{S}_n$. The *Riemannian gradient* of f at $p \in \mathcal{S}_n$

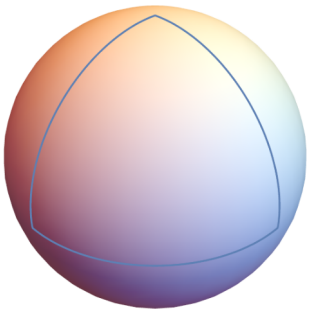


Figure 3.1: Sphere map for the 2-dimensional simplex $\mathcal{S}_2$ (cf. [8, Fig. 2]). The triangle encloses the image $\psi(\mathcal{S}_2) \subset 2\mathbb{S}^2$ of the probability simplex $\mathcal{S}_2$ under the sphere map (3.5).

is given by

$$\operatorname{grad}_{\mathcal{S}_n} f(p) = p\big(\nabla f(p) - \langle p, \nabla f(p)\rangle \mathbb{1}\big). \tag{3.6}$$

By using the so-called *replicator operator*, given by the linear map

$$R_p \colon \mathbb{R}^n \to T_n, \quad x \mapsto R_p[x] := \big(\operatorname{Diag}(p) - pp^\top\big)\, x, \quad p \in \mathcal{S}_n, \tag{3.7}$$

the Riemannian gradient (3.6) can be written as

$$\operatorname{grad}_{\mathcal{S}_n} f(p) = R_p\big[\nabla f(p)\big]. \tag{3.8}$$

Remark 3.1 (Domain of R_p) The replicator operator R_p is turned into an *isomorphism* by restricting the domain to the tangent space T_n

$$R_p \colon T_n \to T_n, \qquad x \mapsto R_p[x] = \big(\operatorname{Diag}(p) - pp^\top\big)\, x, \qquad p \in \mathcal{S}_n, \tag{3.9a}$$
$$R_p^{-1} \colon T_n \to T_n, \qquad x \mapsto R_p^{-1}[x] = \Pi_{T_n} \operatorname{Diag}\big(\tfrac{1}{p}\big) x, \qquad p \in \mathcal{S}_n. \tag{3.9b}$$

In addition, the operator satisfies the relation

$$R_p = R_p \Pi_{T_n} = \Pi_{T_n} R_p, \tag{3.10}$$

where Π_{T_n} denotes the orthogonal projection (3.3) onto the tangent space T_n. △

The exponential map of the *Riemannian (Levi-Civita) connection* on $\mathcal{S}_n$ is as follows.

Proposition 3.1 (Geodesic and Exponential map on $\mathcal{S}_n$; [8, Proposition 2])
The *Riemannian geodesic* is given by

$$\gamma_v(t) = \frac{1}{2}\left(p + \frac{v_p^2}{\|v_p\|^2}\right) + \frac{1}{2}\left(p - \frac{v_p^2}{\|v_p\|^2}\right)\cos\left(\|v_p\|\, t\right) + \frac{v_p}{\|v_p\|}\sqrt{p}\sin\left(\|v_p\|\, t\right) \tag{3.11a}$$

with $t = 1, v_p = v/\sqrt{p}, \gamma_v(0) = p, \dot{\gamma}_v(0) = v$ and

$$V_p = \left\{v \in T_p\mathcal{S}_n : \gamma_v(t) \in \mathcal{S}_n, t \in [0,1]\right\}. \tag{3.11b}$$

The corresponding *exponential map* reads

$$\mathrm{Exp}_p \colon V_p \to \mathcal{S}_n, \quad v \mapsto \mathrm{Exp}_p(v) = \gamma_v(1), \quad p \in \mathcal{S}_n. \tag{3.11c}$$

△

Adopting the *e-connection* from information geometry [3, Section 2.3], [9], the exponential map based on the corresponding affine geodesics reads

$$\mathrm{Exp}_p^e \colon \mathcal{S}_n \times T_n \to \mathcal{S}_n, \qquad (p,v) \mapsto \mathrm{Exp}_p^e(v) = \frac{pe^{\frac{v}{p}}}{\langle p, e^{\frac{v}{p}}\rangle}, \tag{3.12a}$$

$$\mathrm{Exp}_p^{e,-1} \colon \mathcal{S}_n \times \mathcal{S}_n \to T_n, \qquad (p,q) \mapsto \mathrm{Exp}_p^{e,-1}(q) = R_p \log\frac{q}{p}. \tag{3.12b}$$

Specifically, we define

$$\exp_p \colon \mathcal{S}_n \times T_n \to \mathcal{S}_n, \qquad (p,v) \mapsto \mathrm{Exp}_p^e \circ R_p(v) = \frac{pe^{v}}{\langle p, e^{v}\rangle}, \tag{3.13a}$$

$$\exp_p^{-1} \colon \mathcal{S}_n \times \mathcal{S}_n \to T_n, \qquad (p,q) \mapsto \exp_p^{-1}(q) = \Pi_{T_n} \log\frac{q}{p}. \tag{3.13b}$$

The $\exp_p$-map has the following properties.

Lemma 3.2
For $p, q \in \mathcal{S}_n$ and $x, y \in \mathbb{R}^n$ we have

$$\exp_p(x+y) = \exp_{\exp_p(x)}(y) \tag{3.14a}$$

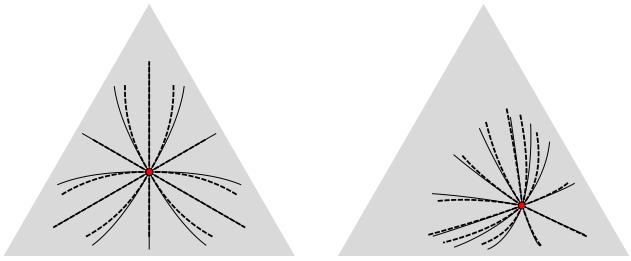


Figure 3.2: First-order approximation of geodesic $\gamma_v(t)$ (cf. [8, Fig. 5]). The *solid lines* display various geodesics $\gamma_{v_i}(t), i \in [k], t \in [t, t_{\max}]$, given by (3.11a), emanating from p (*red points*) with the same speed $\|v_i\|_p = \|v_j\|_p, \forall i, j \in [k]$. The *dashed lines* display the approximations $\mathrm{Exp}^e_p(v_i t), i \in [k], t \in [t, t_{\max}]$ satisfying (3.15).

and the differentials are given by

$$d_q \exp_p(q)[v] = R_{\exp_p(q)}[v] \tag{3.14b}$$

$$d_p \exp_p^{-1}(q)[v] = \Pi_{T_n} \frac{v}{q} \tag{3.14c}$$

The application of the map $\exp_p$ to a vector in $\mathbb{R}^n = T_n \oplus \mathbb{R}\mathbb{1}_n$ does not depend on the $\mathbb{R}\mathbb{1}_n$ component of the argument due to (3.10).

Proof See Appendix A.2. □

Remark 3.2 Since the mappings Exp^e_p and $\exp_p$ do not correspond to the Riemannian connection, the maps (3.12a) and (3.13a) are not length-minimizing with respect to the Riemannian structure. Nevertheless, they provide locally a close approximation (summarized shortly in Proposition 3.3 & illustrated by Fig. 3.2) and are more convenient for numerical computations. △

Proposition 3.3 (First-order approximation of geodesic $\gamma_v(t)$; [8, Proposition 3]) The exponential map of the *e-connection* $\mathrm{Exp}^e_p(vt)$ given by (3.12a) provides locally a *first-order approximation* of the geodesic $\gamma_v(t)$ from (3.11a)

$$\|\gamma_v(t) - \mathrm{Exp}^e_p(vt)\| = \mathcal{O}(t^2). \tag{3.15}$$

△

Next, we define the *Riemannian mean* (also known as *Karcher mean* or *Fréchet mean*) of a given set of points.

Definition 3.4 (Riemannian mean)
The *Riemannian mean* $\bar{p}$ of a set of points $\{p_i\}_{i\in[N]} \subset \mathcal{S}_n$ with weights $\omega \in \Delta_N$ is defined as the minimizer of the objective function

$$\frac{1}{2}\sum_{i\in[N]} \omega_i d^2_{\mathcal{S}_n}(p, p_i), \qquad \text{with} \quad d_{\mathcal{S}_n}(p,q) = 2\arccos\Big(\sum_{i\in[n]} \sqrt{p_i q_i}\Big) \in [0,\pi). \tag{3.16a}$$

The mean $\bar{p}$ satisfies the optimality condition

$$\sum_{i\in[N]} \omega_i \mathrm{Exp}^{-1}_{\bar{p}}(p_i) = 0, \tag{3.16b}$$

where $\mathrm{Exp}^{-1}_{\bar{p}} \colon \mathcal{S}_n \to T_p\mathcal{S}_n$ is the inverse of the exponential map (3.11c). We also use the notation

$$\mathrm{mean}_{\mathcal{S}_n,\omega}(\mathcal{P}), \quad \omega \in \Delta_{N-1}, \quad \mathcal{P} = \{p_1, \ldots, p_N\}, \tag{3.16c}$$

for the *Riemannian mean*. △

Lemma 3.5 ([8, Lemma 3])
The *Riemannian mean* (3.16c) defined as the minimizer of (3.16a) is unique for any data $\mathcal{P} = \{p_i\}_{i\in[N]} \subset \mathcal{S}_n$ and weights $\omega \in \Delta_N$. △

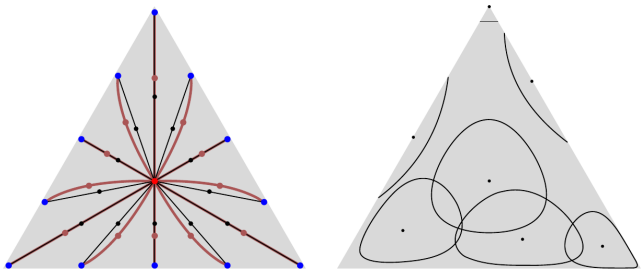
In view of Remark 3.2, we can approximate the Riemannian mean (3.16c) by replacing Exp^{-1}_p (exponential map of *Riemannian connection* (3.11c)) with $\mathrm{Exp}^{e,-1}_p$ (exponential map of *e-connection* (3.12b)). The advantage of using $\mathrm{Exp}^{e,-1}_p$ lies in the fact that we can obtain the following simple closed-form solution of the *approximated mean* based on (3.16b).

Lemma 3.6 (Approximation of the Riemannian mean; [8, Lemma 5])
Let $\mathcal{P} = \{p_1, \ldots, p_N\}$ be a given set of points and $\omega \in \Delta_N$ corresponding weights. Then, by replacing the exponential map Exp^{-1}_p in (3.16b) with $\mathrm{Exp}^{e,-1}_p$, given by (3.12b), yields the following *approximation* of the *Riemannian mean* (3.16c)

$$\mathrm{mean}_{\mathcal{S}_n,\omega}(\mathcal{P}) \approx \exp_{\mathbb{1}_{\mathcal{S}_n}}\left(\sum_{i\in[N]} \omega_i \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(p_i)\right) = \frac{\mathrm{mean}_{g,\omega}(\mathcal{P})}{\langle \mathbb{1}, \mathrm{mean}_{g,\omega}(\mathcal{P})\rangle}, \tag{3.17a}$$

where $\mathrm{mean}_{g,\omega}(\mathcal{P})$ denotes the *weighted geometric mean*

$$\mathrm{mean}_{g,\omega}(\mathcal{P}) = \prod_{i\in[N]} p_i^{\omega_i}. \tag{3.17b}$$



(a) (b)

Figure 3.3: Geometry of the probability simplex induced by the Fisher–Rao metric. (cf. [8, Fig. 3]) (a) shows the geodesics from the barycenter *(red)* to various points *(blue)*. The *black lines* are Euclidean geodesics and the *brown lines* are non-Euclidean geodesics. The points along these lines correspond to the *Euclidean* and *Riemannian mean*, respectively. **(b)** The *contour lines* denote the points which have the same Riemannian distance from the respective center point *(black dots)*.

Proof See Appendix A.2. □

This lemma enables us to approximate the *Riemannian mean* with a closed-form expression, namely the *normalized weighted geometric mean* (3.17a). This result will be extremely useful for numerical computations (Section 3.3).

3.1.2 Global Object: Assignment Manifold

The *main* mathematical object of this thesis is the so-called *assignment manifold*, given by the product manifold

$$\mathcal{W} := \underbrace{\mathcal{S}_n \times \cdots \times \mathcal{S}_n}_{m\text{-times}} \tag{3.18}$$

with constant *tangent space*

$$\mathcal{T}_{\mathcal{W}} := \underbrace{T_n \times \cdots \times T_n}_{m\text{-times}} \tag{3.19}$$

and Riemannian structure $(\mathcal{W}, g^{\mathcal{W}})$ given by the *Riemann product metric*

$$g_W^{\mathcal{W}} : \mathcal{T}_{\mathcal{W}} \times \mathcal{T}_{\mathcal{W}} \to \mathbb{R}, \quad (U,V) \mapsto g_W^{\mathcal{W}}(U,V) := \sum_{k=1}^{m} g_{W_k}^{\mathcal{S}_n}(U_k, V_k), \quad \forall\, U, V \in \mathcal{T}_{\mathcal{W}}. \tag{3.20}$$

We identify $\mathcal{W}$ with the embedding into $\mathbb{R}^{m\times n}$

$$\mathcal{W} = \{W \in \mathbb{R}^{m\times n} \colon W\mathbb{1}_n = \mathbb{1}_m \text{ and } W_{ij} > 0 \text{ for all } i \in [m], j \in [n]\}. \tag{3.21}$$

In words, a point $W \in \mathcal{W}$ is a row-stochastic matrix $W \in \mathbb{R}^{m\times n}$ representing *global* label assignments on the whole set of nodes $\mathcal{V}$. Each row vector $W_i \in \mathcal{S}_n$ lives on the probability simplex $\mathcal{S}_n$ and represents *local* label assignments for every node $i \in \mathcal{V}$. Due to this embedding of $\mathcal{W}$, the tangent space $\mathcal{T}_{\mathcal{W}}$ can be identified with

$$\mathcal{T}_{\mathcal{W}} = \{V \in \mathbb{R}^{m\times n} \colon V\mathbb{1}_n = 0\}. \tag{3.22}$$

Therefore, every row vector V_i is contained in T_n for every $i \in \mathcal{V}$. The global uniform distribution, given by the uniform distribution in every row, again called *barycenter*, is denoted by

$$\mathbb{1}_{\mathcal{W}} := (\mathbb{1}_{\mathcal{S}_n}, \ldots, \mathbb{1}_{\mathcal{S}_n}) = \mathbb{1}_m \mathbb{1}_{\mathcal{S}_n}{}^{\top} \in \mathcal{W}, \tag{3.23}$$

where the second equality is due to the embedding (3.21).

The maps and operators defined on the probability simplex $\mathcal{S}_n$, have naturally extensions on the product manifold $\mathcal{W}$. The *orthogonal projection* onto $\mathcal{T}_{\mathcal{W}}$ is given by

$$\Pi_{\mathcal{T}_{\mathcal{W}}} \colon \mathbb{R}^{m\times n} \to \mathcal{T}_{\mathcal{W}}, \quad X \mapsto \Pi_{\mathcal{T}_{\mathcal{W}}}[X] = \begin{pmatrix} \Pi_{T_n}[X_1] \\ \vdots \\ \Pi_{T_n}[X_m] \end{pmatrix}, \tag{3.24}$$

where Π_{T_n} is the orthogonal projection (3.3) onto T_n. The *replicator operator* is given by

$$R_W \colon \mathbb{R}^{m\times n} \to \mathcal{T}_{\mathcal{W}}, \quad X \mapsto R_W[X] = \begin{pmatrix} R_{W_1}[X_1] \\ \vdots \\ R_{W_m}[X_m] \end{pmatrix}, \quad W \in \mathcal{W}, \tag{3.25}$$

where R_{W_i} is the replicator operator (3.7). The *lifting map* is defined by

$$\exp_W(V) \colon \mathcal{W} \times \mathcal{T}_{\mathcal{W}} \to \mathcal{W}, \quad (W, V) \mapsto \exp_W(V) = \begin{pmatrix} \exp_{W_1}(V_1) \\ \vdots \\ \exp_{W_m}(V_m) \end{pmatrix}, \tag{3.26}$$

where W_i, V_i for $i \in \mathcal{V}$ are the row vectors of matrices W, V, respectively, and $\exp_{W_i}(V_i)$ is the lifting map (3.13a). The mappings $\exp_W^{-1}, \mathrm{Exp}_W^e, \mathrm{Exp}_W^{e,-1}$ are similarly defined based on (3.13b), (3.12a) and (3.12b).

Due to (3.8), the *Riemannian gradient* of a smooth function $f\colon \mathcal{W} \to \mathbb{R}$ is given by

$$\operatorname{grad} f(W) = R_W[\nabla f(W)] \quad \text{for } W \in \mathcal{W}. \tag{3.27}$$

3.2 Vector Fields on the Assignment Manifold

In this section we summarize the main idea of [65]: Any given vector field $X \in \mathfrak{X}(\mathcal{W})$ on $\mathcal{W}$ can be *transformed* onto the tangent space $\mathcal{T}_\mathcal{W}$. The advantage of using this transformation lies in the fact that $\mathcal{T}_\mathcal{W} \subset \mathbb{R}^{m\times n}$ is a vector space of matrices, where established numerical integration methods can be applied.

The next theorem states the main result for general vector fields $X \in \mathfrak{X}(\mathcal{W})$.

Theorem 3.7 (Transformation of Vector Fields on $\mathcal{W}$)
Let $\exp_{\mathbb{1}_\mathcal{W}}\colon \mathcal{T}_\mathcal{W} \to \mathcal{W}$ be the lifting map (3.26) at the barycenter $\mathbb{1}_\mathcal{W} := \mathbb{1}_m \mathbb{1}_{\mathcal{S}_n}{}^\top \in \mathcal{W}$, $X \in \mathfrak{X}(\mathcal{W})$ be a given vector field, and $J \subset \mathbb{R}$ be an open interval with $0 \in J$. Then, a curve $W\colon J \to \mathcal{W}$ solves (3.28a) if and only if the curve $V\colon J \to \mathcal{T}_\mathcal{W}$ with $W(t) = \exp_{\mathbb{1}_\mathcal{W}} \circ V(t)$ solves (3.28b).

$$\dot W(t) = X_{W(t)}, \tag{3.28a}$$

$$\dot V(t) = R^{-1}_{\exp_{\mathbb{1}_\mathcal{W}}(V(t))}\left[X_{\exp_{\mathbb{1}_\mathcal{W}}(V(t))}\right]. \tag{3.28b}$$

Proof See Appendix A.2. □

The above setting works for general vector fields and metrics on the assignment manifold $\mathcal{W}$. In the remainder of this section we restrict this setting to *Riemannian gradients* and the *Fisher–Rao metric* (3.20). Suppose $f\colon \mathcal{W} \to \mathbb{R}$ is a general smooth objective function given on the assignment manifold $\mathcal{W}$. Our strategy is to minimize this function by following the *Riemannian gradient descent flow*

$$\dot W(t) = -\operatorname{grad}_\mathcal{W} f(W(t)). \tag{3.29}$$

Applying Theorem 3.7 with $X_{W(t)} = -\operatorname{grad}_\mathcal{W} f(W(t))$ transforms (3.29) onto the tangent space $\mathcal{T}_\mathcal{W}$. This transformation is summarized by the following corollary.

Corollary 3.8 (Transformation of Riemannian Gradient Flows on $\mathcal{W}$)
Let $\exp_{\mathbb{1}_\mathcal{W}} \colon T^m \to \mathcal{W}$ be the lifting map (3.26) at the barycenter $\mathbb{1}_\mathcal{W} := \mathbb{1}_m \mathbb{1}_{\mathcal{S}_n}{}^\top \in \mathcal{W}$, $\operatorname{grad}_\mathcal{W} f(W(t))$ the *Riemannian gradient* of a general smooth objective function $f \colon \mathcal{W} \to \mathbb{R}$, and the manifold $\mathcal{W}$ is equipped with the *Fisher–Rao metric* (3.20). Then, the integral curves $W(t), V(t)$ of the following gradient flows

$$\dot{W}(t) = -\operatorname{grad}_\mathcal{W} f(W(t)), \quad W(0) = \mathbb{1}_\mathcal{W}, \tag{3.30a}$$

$$\dot{V}(t) = -\nabla f(\exp_{\mathbb{1}_\mathcal{W}}(V(t))), \quad V(0) = 0_{m\times n}. \tag{3.30b}$$

can be transformed into each other via the lifting map

$$W(t) = \exp_{\mathbb{1}_\mathcal{W}}\big(V(t)\big). \tag{3.31}$$

Proof See Appendix A.2. □

3.3 Image Labeling on the Assignment Manifold

A given image can be modeled by an undirected graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with $m := |\mathcal{V}|$ vertices. Let

$$f \colon \mathcal{V} \to \mathcal{F}, \quad i \mapsto f_i \in \mathcal{F}, \quad f(\mathcal{V}) =: \mathcal{F}_\mathcal{V} \subset \mathcal{F} \tag{3.32}$$

be data on the graph given in a metric space $(\mathcal{F}, d)$. We call $\mathcal{F}_\mathcal{V}$ *image data* given by *features* f_i extracted from a raw image at pixel $i \in \mathcal{V}$. Along with f we assume prototypical data

$$\mathcal{X} = \{\ell_1, \ldots, \ell_n\} \subset \mathcal{F} \tag{3.33}$$

which we call *labels*. Assume a suitable distance function

$$d_\mathcal{F} \colon \mathcal{F} \times \mathcal{X} \to \mathbb{R}, \tag{3.34}$$

is given which measures the similarity between features and labels. The *image labeling problem* consists of finding an assignment $\mathcal{V} \to \mathcal{X}$ that assigns class labels to nodes depending on the image data $\mathcal{F}_\mathcal{V}$ and the local context encoded by the graph structure $\mathcal{G}$. These global assignments are modeled as points $W \in \mathcal{W}$ on the *assignment manifold* $\mathcal{W}$.

More precisely, we are interested in the *posterior probability* of assigning label ℓ_j to vertex i given the datum f_i, which is given by the element

$$W_{ij} = \Pr(\ell_j | f_i), \quad i \in [m], \quad j \in [n], \tag{3.35}$$

of the assignment matrix W. Thus, the assignments of to each vertex $i \in \mathcal{V}$ is represented by the row vector

$$W_i \in \mathcal{S}_n, \quad i \in \mathcal{V}. \tag{3.36}$$

$\mathcal{G}$ may be a grid graph (with self-loops) as in low-level image processing or a less structured graph, with arbitrary connectivity in terms of the neighborhoods

$$\mathcal{N}_i = \{k \in \mathcal{V} \colon ik = ki \in \mathcal{E}\} \cup \{i\}, \quad i \in \mathcal{V}, \tag{3.37}$$

where ik is a shorthand for the undirected edge $\{i,k\} \in \mathcal{E}$. We require these neighborhoods to satisfy the relations

$$k \in \mathcal{N}_i \quad \Leftrightarrow \quad i \in \mathcal{N}_k, \qquad \forall i,k \in \mathcal{V}. \tag{3.38}$$

We associate with each neighborhood $\mathcal{N}_i$ from (3.37) weights $\omega_{ik} \in \mathbb{R}$ for all $k \in \mathcal{N}_i$, satisfying

$$\omega_{ik} > 0 \quad \text{and} \quad \sum_{k \in \mathcal{N}_i} \omega_{ik} = 1, \quad \text{for all } i \in \mathcal{V}. \tag{3.39}$$

These weights parametrize the regularization property of the assignment flow below.

3.3.1 Assignment Flow

Based on the given data (3.32) and labels (3.33), we define the *distance matrix*

$$D \in \mathbb{R}^{m \times n}, \quad D_{i,j} := d_{\mathcal{F}}(f_i, \ell_j), \quad i \in [m], j \in [n]. \tag{3.40}$$

This distance information is *lifted* onto the manifold $\mathcal{W}$ by the *likelihood matrix*

$$L = L(W) \in \mathcal{W}, \quad L(W) := \exp_W\left(-D/\rho\right), \quad \rho > 0, \tag{3.41}$$

where ρ is a *scaling parameter* for the distance matrix D, and $\exp_W$ is the *lifting map* (3.26). This representation of the data is regularized by the *approximation of the Riemannian mean* (3.17a) in the local neighborhoods (3.37) using the weights (3.39).

Thus, the *similarity matrix* is given by

$$S = S(W) \in \mathcal{W}, \quad S_i(W) := \exp_{\mathbb{1}_{\mathcal{S}_n}}\Big(\sum_{k\in\mathcal{N}_i} w_{ik} \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(L_k(W_k))\Big), \quad i \in \mathcal{V}, \tag{3.42}$$

where $\mathcal{N}(i)$ is the local neighborhood (3.37).

The similarity matrix induces the *assignment flow* through a system of spatially coupled nonlinear ODEs which evolves the assignment vectors

$$\dot{W}(t) = R_{W(t)}\big[S(W(t))\big] = W(t)\left(S(W(t)) - \langle W, S(W(t))\rangle \mathbb{1}\right), \tag{3.43a}$$
$$W(0) = \mathbb{1}_{\mathcal{W}} \in \mathcal{W}, \tag{3.43b}$$

where $\mathbb{1}_{\mathcal{W}}$ denotes the global uniform distribution, given by (3.23). Integrating this flow numerically yields curves $W_i(t) \in \mathcal{S}_n$ for every pixel $i \in \mathcal{V}$ that emanate from $W_i(0) = \mathbb{1}_{\mathcal{S}_n}$ and approach some vertex (unit vector) of $\overline{\mathcal{S}_n} = \Delta_n$. Hence, a unique label assignment is obtained after a trivial rounding $W_i(t)$ for sufficiently large $t > 0$. The overall geometric approach is summarized and illustrated by Fig. 3.4.

3.3.2 Numerical Integration of the Flow

Theorem 3.7 shows how a given vector field on $\mathcal{W}$ can be transformed onto the tangent space $\mathcal{T}_{\mathcal{W}}$ via the lifting map $\exp_{\mathbb{1}_{\mathcal{W}}}$. Accordingly, for the *assignment flow* (3.43) this transformation is as follows. The *integral curves* of the following ODEs

$$\dot{W}(t) = R_{W(t)}\big[S(W(t))\big], \quad W(0) = \mathbb{1}_{\mathcal{W}} \in \mathcal{W}, \tag{3.44a}$$
$$\dot{V}(t) = S\big(\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))\big), \quad V(0) = 0 \in \mathcal{T}_{\mathcal{W}}, \tag{3.44b}$$

are equivalent via the parametrization $W(t) = \exp_{\mathbb{1}_{\mathcal{W}}}(V(t))$. The flow (3.44b) purely evolves on the vector space $\mathcal{T}_{\mathcal{W}}$, where standard Runge-Kutta methods (cf. Section 2.3.2) can be used for numerical integration.

In general, for an arbitrary ODE

$$\dot{V}(t) = F\big(W(t)\big) = F\big(\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))\big), \quad V(0) = 0, \tag{3.45}$$

evolving on the vector space $\mathcal{T}_{\mathcal{W}}$, we mainly use *explicit Runge–Kutta methods* (2.58) for the numerical integration, i.e.

$$V^{(n+1)} = V^{(n)} + h^{(n)} \sum_{i=1}^{s} b_i k_{n,i}, \quad V^{(0)} = 0, \tag{3.46}$$

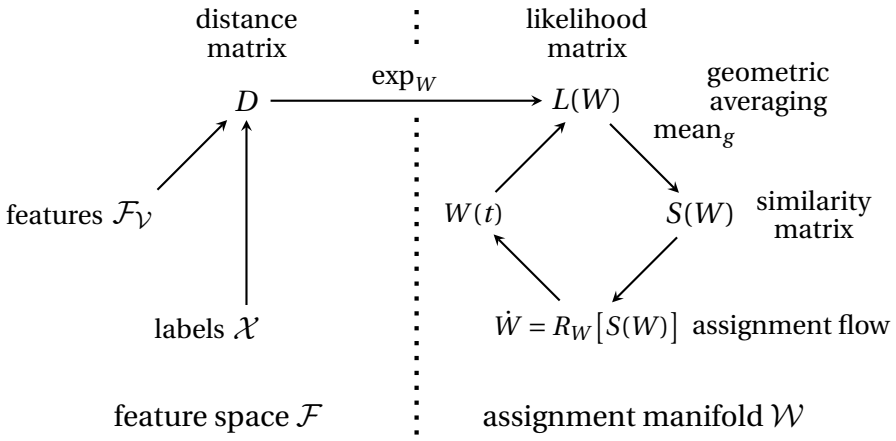


Figure 3.4: Overview of the geometric approach [8]. In a feature space $\mathcal{F}$, the distance D between given data $\mathcal{F}_\mathcal{V}$ and labels $\mathcal{X}$ is computed by a suitable distance measure. These information is lifted to the *assignment manifold* $\mathcal{W}$ which gives the likelihood $L(W)$. In order to obtain spatial coherent assignments, the similarity matrix $S(W)$ is computed by geometric averaging over spatial neighborhoods. The resulting inference corresponds to following the *assignment flow*, that is a replicator dynamic which is induced from $S(W)$.

with step-size $h^{(n)} \in \mathbb{R}_{>0}$. The stages $k_{n,i}$ depend on the right hand side F of (3.45), and are defined in (2.58a)-(2.58c). By using the parametrization $W = \exp_{\mathbb{1}_\mathcal{W}}(V)$, this update scheme translates to a *multiplicative update formula* on $\mathcal{W}$

$$W^{(n+1)} = \exp_{\mathbb{1}_\mathcal{W}}\left(V^{(n+1)}\right) \overset{(3.46)}{=} \exp_{\mathbb{1}_\mathcal{W}}\left(V^{(n)} + h^{(n)} \sum_{i=1}^{s} b_i k_{n,i}\right) \tag{3.47a}$$

$$\overset{(3.14a)}{=} \exp_{W^{(n)}}\left(h^{(n)} \sum_{i=1}^{s} b_i k_{n,i}\right) = \frac{W^{(n)} e^{h^{(n)} \sum_{i=1}^{s} b_i k_{n,i}}}{\left\langle W^{(n)}, e^{h^{(n)} \sum_{i=1}^{s} b_i k_{n,i}} \right\rangle}, \tag{3.47b}$$

with initial value $W^{(0)} = \mathbb{1}_\mathcal{W}$.

Remark 3.3 The construction of the *geometric* Runge-Kutta schemes (3.47b) can be viewed from a different perspective. Setting $\Lambda(V, W) := \exp_W(V)$ gives an Lie-group action $\Lambda\colon \mathcal{T}_\mathcal{W} \times \mathcal{W} \to \mathcal{W}$ of the vector space $\mathcal{T}_\mathcal{W}$ viewed as an additive group on the

assignment manifold $\mathcal{W}$. In [81] this action is used to numerically integrate the assignment flow by applying geometric Runge-Kutta methods (3.47b). △

Using the explicit Runge–Kutta method on $\mathcal{T}_{\mathcal{W}}$ gives the *geometric explicit Euler update* on $\mathcal{W}$

$$W^{(n+1)} = \frac{W^{(n)} e^{h^{(n)} F(W^{(n)})}}{\left\langle W^{(n)}, e^{h^{(n)} F(W^{(n)})} \right\rangle}, \quad W^{(0)} = \mathbb{1}_{\mathcal{W}} \in \mathcal{W}. \tag{3.48}$$

Definition 3.9 (Linear assignment flow)
The *linear assignment flow* approximates the mapping (3.42) as part of the assignment flow (3.43) by

$$\dot{W} = R_W \Big[S(W_0) + dS(W_0) \big[\exp_{\mathbb{1}_{\mathcal{W}}}^{-1}(W) \big] \Big], \quad W_0 = W(0) = \mathbb{1}_{\mathcal{W}} \in \mathcal{W}. \tag{3.49}$$

Although, this flow is still *nonlinear* on $\mathcal{W}$, its transformed flow (Theorem 3.7)

$$\dot{V} = S(W_0) + dS(W_0)[V], \quad V(0) = 0, \tag{3.50}$$

is *linear* and defined on the vector space $\mathcal{T}_{\mathcal{W}}$. △

The differential $dS(W)[V]$ of the similarity map is stated in the following lemma.

Lemma 3.10
The i-th component of the differential $dS(W) \colon \mathcal{T}_{\mathcal{W}} \to \mathcal{T}_{\mathcal{W}}$ is given by

$$dS_i(W) \colon \mathcal{T}_{\mathcal{W}} \to T_n, \quad dS_i(W)[V] = \sum_{k \in \mathcal{N}_i} \omega_{ik} R_{S_i(W)} \left[\frac{V_k}{W_k} \right], \tag{3.51}$$

for all $V \in \mathcal{T}_{\mathcal{W}}$ and $i \in \mathcal{V}$.

Proof See Appendix A.2. □

Remark 3.4 (Linearity of (3.50) with respect to parameter) By fixing $S(W_0)$, the right hand side of (3.50) is *linear* with respect to both the tangent vector V and the parameters ω_{ik} in the differential $dS(W_0)$ (see (3.51)), that makes this approach attractive for parameter estimation (investigated in Chapter 5). △

Remark 3.5 (Parametrization using Exp_p^e) The work of [81] introduced the *linear assignment flow* by using the exponential map Exp_p^e, given by (3.12a), with respect to

the *e-connection*

$$\dot{W} = R_W\Big[S(W_0) + dS(W_0)\big[\operatorname{Exp}^{e,-1}_{\mathbb{1}_{\mathcal{W}}}(W)\big]\Big], \quad W_0 = W(0) = \mathbb{1}_{\mathcal{W}} \in \mathcal{W}, \tag{3.52}$$

and with parametrization [81, Prop. 4.2]

$$W(t) = \operatorname{Exp}^{e}_{W_0}(V(t)), \quad \dot{V} = R_{W_0}\left[S(W_0) + dS(W_0)\left[V\right]\right], \quad V(0) = 0 \tag{3.53}$$

It can be shown that the parametrization (3.49) and (3.53) differ only by a factor. △

3.4 Experiments

3.4.1 Implementation Details

Assignment Normalization

Each vector W_i approaches some vertex e_i of the simplex and thus some entries of W_i converge to zero. However, due to our optimization scheme every vector W_i evolves on the interior of the simplex $\mathcal{S}$, that is, all entries of W_i have to be positive all the time. Since there is a limit for the precision of representing small positive numbers on a computer, we avoid numerical problems by adopting the normalization strategy of [8]. After each iteration, we check all W_i and whenever an entry drops below $\varepsilon = 10^{-10}$, we rectify W_i by

$$W_i \leftarrow \frac{1}{\langle \mathbb{1}, \tilde{W}_i \rangle} \tilde{W}_i\,, \quad \tilde{W}_i = W_i - \min_{j=1,\ldots,n}\{W_{i,j}\} + \varepsilon\,, \quad \varepsilon = 10^{-10}\,. \tag{3.54}$$

Thus, the constant ε plays the role of 0 in our implementation. Our numerical experiments showed that this operation avoids numerical issues.

Termination Criterion

In all experiments, the normalized averaged entropy

$$\frac{1}{m\log(n)} H(W) = -\frac{1}{m\log(n)} \sum_{i\in\mathcal{V}} \sum_{k=1}^{n} W_{i,k} \log\big(W_{i,k}\big), \quad \text{for} \quad W \in \mathcal{W}, \tag{3.55}$$

was used as a termination criterion, i.e. if the value drops below a certain threshold the algorithm is terminated. Due to this normalization, the value does not depend on the number of labels and thus the threshold is comparable across different models with a varying number of pixels and labels.

For example, a threshold of 10^{-3} means in practice that, up to a small fraction of nodes $i \in \mathcal{V}$, all rows W_i of the assignment matrix W are very close to unit vectors and thus indicate an almost unique assignment of the prototypes or labels to the observed data.

3.4.2 Parameter Influence

In this section we illustrate the parameter influence of the scaling parameter ρ and the spatial scale $|\mathcal{N}|$ on the assignment (cf. Fig. 3.5). By using *uniform weights* for regularization, i.e. $\omega_i = \frac{1}{|\mathcal{N}_i|}\mathbb{1}_{|\mathcal{N}_i|}$ for every node $i \in \mathcal{V}$, the similarity matrix (3.42) simplifies to the *normalized geometric mean* (3.17b).

The task is to label a RGB-image $f\colon \mathcal{V} \to [0,1]^3$ on the grid graph $\mathcal{G} = (\mathcal{V},\mathcal{E})$ with different neighborhood size $|\mathcal{N}(i)| \in \{3\times 3, 5\times 5, 7\times 7\}$, $i \in \mathcal{V}$. Prototypical colors $\mathcal{X} = \{\ell_1,\dots,\ell_{12}\} \subset [0,1]^3$ serve as labels (cf. Fig. 3.5, top right). The distance matrix is computed by using the $\|\cdot\|_1$ norm and a scaling factor $\rho > 0$ by

$$D_i = \frac{1}{\rho}\big(\|f(i)-\ell_1\|_1,\dots,\|f(i)-\ell_{12}\|_1\big), \quad i \in \mathcal{V}. \tag{3.56}$$

We use a constant step-size of $h = 1$ for the numerical integration of the assignment flow (3.44b), and set the threshold for the normalized average entropy termination criterion (3.55) to 10^{-3}.

Fig. 3.5 shows the influence of the *scaling parameter* ρ and the *spatial scale* $|\mathcal{N}|$ on the assignment. Increasing the strength of the data (smaller ρ) leads to a faster decrease in entropy (cf. Fig. 3.6) and therefore to an earlier convergence of the process to a specific labeling. Thus, a stronger weighted data term yields a less regularized result due to the rapid decision for a labeling at an early stage of the algorithm.

3.5 Extensions

In this section we briefly list several *extensions* of the geometric approach explained in this chapter. For more details we refer the reader to the respective references.

Mathematical aspects [63, 64]. In [63] a variational formulation of the assignment flow is studied leading to an natural extension from graphs to the continuous domain in the "zero-scale limit". The work of [64] presents a more classical additive variational reformulation related to the continuous cut approach, using Riemannian distances induced by the Fisher–Rao metric for regularization.

Geometric numerical integration [81]. The work presents a comprehensive study of geometric integration techniques for the numerical integration of the assignment flow in a stable, efficient and parameter-free way.

Unsupervised label learning [82, 83]. This work extends the geometric approach to *unsupervised* scenarios where no labels are given. The work determines labels in a completely unsupervised way by data self-assignment. This results in a self-assignment flow which has connections to low-rank matrix factorization and discrete optimal mass transport.

Evaluation of discrete graphical models [6, 36]. This extension introduces a novel approach to *maximum a posteriori (MAP) inference* based on discrete graphical models and the assignment manifold. The idea is to *smoothly approximate* the LP relaxation and restrict it to the assignment manifold. In this work, the integration of the corresponding Riemannian gradient flow and a rounding mechanism to integral solutions is combined. We present and elaborate this approach in Chapter 4 in detail.

Parameter learning [35, 37]. This work studies the *inverse problem* of model parameter learning for pixelwise image labeling. Based on given training data with ground truth, the weights of the *weighted geometric mean* that parametrize the adaptivity of the assignment flow are learned. This is accomplished by a Riemannian gradient flow on the manifold of parameters that determine the regularization properties of the assignment flow. We present and discuss this approach in Chapter 5 in detail.

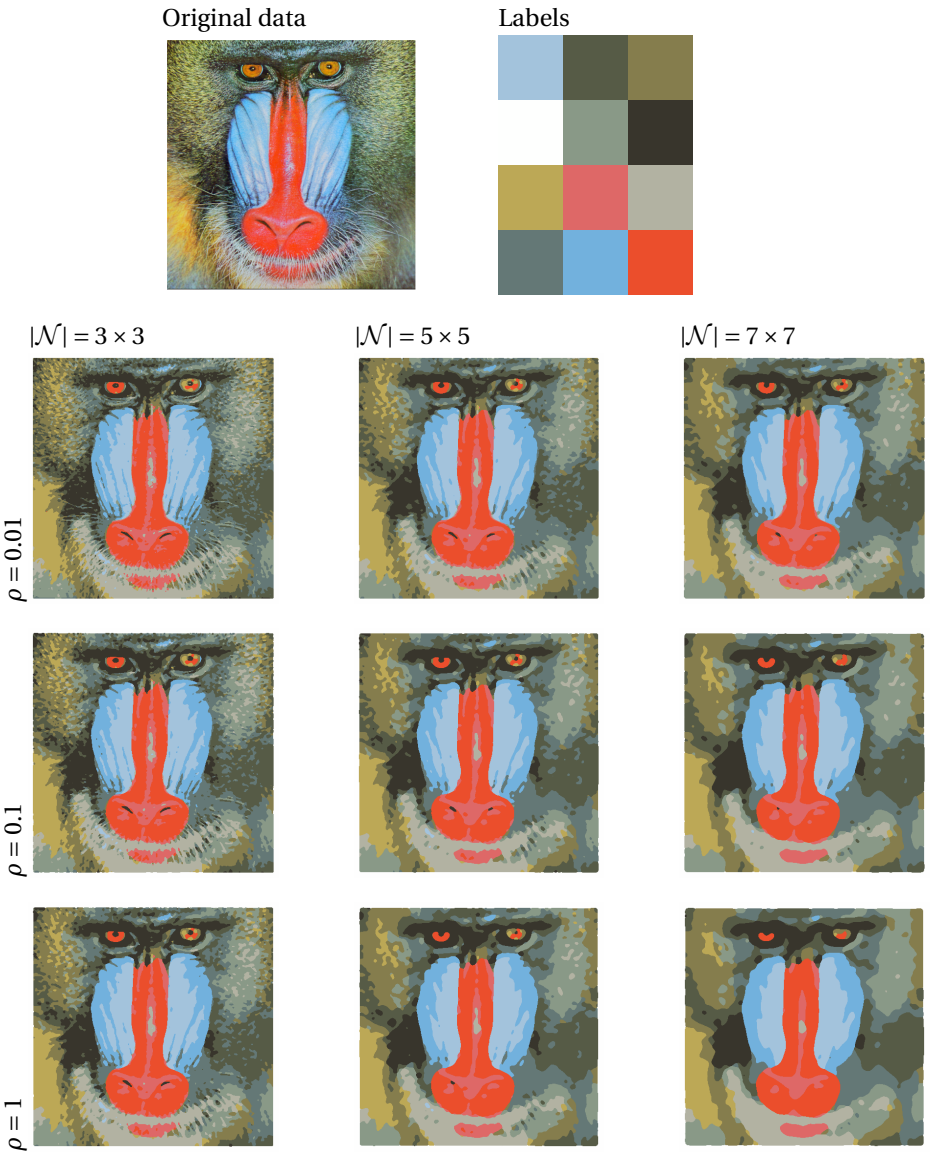


Figure 3.5: Parameter influence of the *scaling parameter* ρ and the *spatial scale* $\mathcal{N}$ on the assignment. This plot shows the parameter influence of ρ and $|\mathcal{N}|$ on the assignment of 12 prototypical labels to the input data. *Uniform weights* are used for regularization, i.e. $\omega_i = \frac{1}{|\mathcal{N}_i|}\mathbb{1}_{|\mathcal{N}_i|}$ for every node $i \in \mathcal{V}$. Increasing size of the spatial scale $|\mathcal{N}|$ leads to more *regularized* labelings. In contrast, a stronger weighted data term (smaller values of ρ) yields a less regularized result due to the rapid decision for a labeling at an early stage of the algorithm (cf. Fig. 3.6).

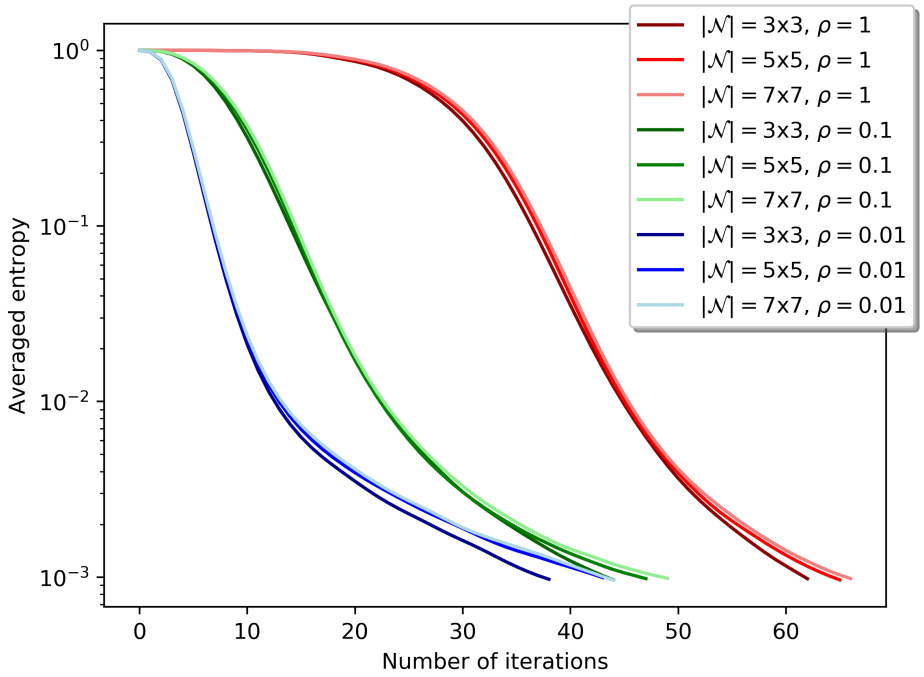


Figure 3.6: Averaged entropy (3.55) **for different values of** ρ **and** $|\mathcal{N}|$**.** This plot shows the corresponding values of the normalized average entropy (3.55) for the experiments of Fig. 3.5. The curves are sorted according to parameter ρ, i.e. $\rho = 1$ (*red curves*), $\rho = 0.1$ (*green curves*) and $\rho = 0.01$ (*blue curves*). By increasing the strength of the data term (smaller values of the scaling parameter ρ), the entropy drops more rapidly and hence converges faster to an integral labeling. A different size of the spatial scale $|\mathcal{N}|$ has no significant influence on the number of iterations.

Chapter 4

Inference based on Graphical Models and Assignment

In this chapter we introduce a novel approach to *maximum a posteriori (MAP) inference* based on discrete graphical models (2.31b) and the assignment manifold (3.18). The main idea is to restrict the LP relaxation (2.42) to the assignment manifold and then smoothly approximate the resulting objective function. Thereby, we combine the integration of the corresponding Riemannian gradient flow, and a rounding mechanism to integral solutions. In order to not disturb the overall line of reasoning all technical proofs are collected as Appendix A.3. If a proof gives insights into a certain formula or structure, we include it in the main text.

This chapter is based on the joint work with Fabrizio Savarino, Judit Recknagel, Freddie Åström and Christoph Schnörr, that was published as a conference paper [6] and as a more elaborated journal version [36].

4.1 Objective Function

Suppose observed image data is modeled by a grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$, $|\mathcal{V}| = m$, and each node $i \in \mathcal{V}$ indexes a pixel location, to which a label from the discrete set

$$x_i \in \mathcal{X} = \{\ell_1, \dots, \ell_n\} \tag{4.1}$$

is assigned. Again, we call this finite set $\mathcal{X}$ *labels* or *prototypes*. Then, the image labeling problem can be formulated in terms of the *discrete energy function*

$$\min_{x \in \mathcal{X}^m} E(x), \qquad E(x) = \sum_{i \in \mathcal{V}} \theta_i(x_i) + \sum_{ij \in \mathcal{E}} \theta_{ij}(x_i, x_j). \tag{4.2}$$

As we have seen in Section 2.2.2, this problem corresponds to *MAP-inference* based on discrete graphical models (2.31b). The variables θ_i denote the given unary potentials and θ_{ij} the given pairwise potentials. If not otherwise specified, we use the following *unary potentials*

$$\theta_i(x_i) \in \left\{ d_{\mathcal{F}}(f_i, \ell_1), \dots, d_{\mathcal{F}}(f_i, \ell_n) \right\}, \quad i \in [m], \tag{4.3}$$

where $d_{\mathcal{F}}$ is a suitable distance function, and as *pairwise potentials* the so-called *Potts prior*

$$\theta_{ij}(x_i, x_j) = \begin{cases} \lambda, & \text{if } x_i \neq x_j \\ 0, & \text{otherwise} \end{cases}, \quad \text{with } \lambda > 0. \tag{4.4}$$

The function $E(x)$ has the format of variational problems comprising a *data term* and a *regularizer*.

4.1.1 Smooth Approximation of the LP Relaxation

In order to conform to the smooth geometric setting explained in Chapter 3, we proceed in two steps: First, we reformulate the LP relaxation (2.42) in terms of the node variables $\mu_{\mathcal{V}}$, and afterwards *smooth* the resulting objective function.

Our first step is summarized in the following Lemma.

Lemma 4.1 (Reformulation of the LP-relaxation (2.42))
The LP-relaxation (2.42) is equivalent to the problem

$$\min_{\mu_{\mathcal{V}} \in \Delta_n^m} \Big(\sum_{i \in \mathcal{V}} \langle \theta_i, \mu_i \rangle + \sum_{ij \in \mathcal{E}} d_{\theta_{ij}}(\mu_i, \mu_j) \Big) \tag{4.5a}$$

involving the *local Wasserstein distances*

$$d_{\theta_{ij}}(\mu_i, \mu_j) := \min_{\mu_{ij} \in \Pi(\mu_i, \mu_j)} \langle \theta_{ij}, \mu_{ij} \rangle \,, \tag{4.5b}$$

defined for every edge $ij \in \mathcal{E}$ with $\Pi(\mu_i, \mu_j)$ due to (2.41b). These distances take the pairwise model parameters θ_{ij} into account. △

Proof The claim follows by reformulating the LP-relaxation based on the local polytope constraints (2.41)

$$\begin{aligned} \min_{\mu \in \mathcal{L}_{\mathcal{G}}} \langle \theta, \mu \rangle &= \min_{\mu \in \mathcal{L}_{\mathcal{G}}} \langle \theta_{\mathcal{V}}, \mu_{\mathcal{V}} \rangle + \langle \theta_{\mathcal{E}}, \mu_{\mathcal{E}} \rangle \\ &= \min_{\mu_{\mathcal{V}} \in \Delta_n^m} \Big(\langle \theta_{\mathcal{V}}, \mu_{\mathcal{V}} \rangle + \min_{\mu_{\mathcal{E}}} \sum_{ij \in \mathcal{E}} \big(\langle \theta_{ij}, \mu_{ij} \rangle + \delta_{\Pi(\mu_i, \mu_j)}(\mu_{ij}) \big) \Big) \\ &= \min_{\mu_{\mathcal{V}} \in \Delta_n^m} \Big(\sum_{i \in \mathcal{V}} \langle \theta_i, \mu_i \rangle + \sum_{ij \in \mathcal{E}} \min_{\mu_{ij} \in \Pi(\mu_i, \mu_j)} \langle \theta_{ij}, \mu_{ij} \rangle \Big) \\ &= \min_{\mu_{\mathcal{V}} \in \Delta_n^m} \Big(\sum_{i \in \mathcal{V}} \langle \theta_i, \mu_i \rangle + \sum_{ij \in \mathcal{E}} d_{\theta_{ij}}(\mu_i, \mu_j) \Big). \end{aligned}$$

□

The function (4.5a) is formulated in terms of $\mu_{\mathcal{V}}$, but it is non-smooth. Therefore, we *smooth* the convex but non-smooth (piecewise-linear (cf. [59, Def. 2.47])) local Wasserstein distances (4.5b) with a general convex *smoothing function* F_τ,

$$d_{\theta_{ij},\tau}(\mu_i,\mu_j) = \min_{\mu_{ij}\in\Pi(\mu_i,\mu_j)} \big\{\langle\theta_{ij},\mu_{ij}\rangle + F_\tau(\mu_{ij})\big\}, \quad ij\in\mathcal{E}, \quad F_\tau\in\mathcal{F}_0, \quad \tau>0, \tag{4.6}$$

with *smoothing parameter* τ. Based on Lemma 4.1 and the regularized local Wasserstein distances (4.6), we study the objective function

$$E_\tau(\mu_{\mathcal{V}}) = \langle\theta_{\mathcal{V}},\mu_{\mathcal{V}}\rangle + \sum_{ij\in\mathcal{E}} d_{\theta_{ij},\tau}(\mu_i,\mu_j), \qquad \tau>0, \tag{4.7}$$

which is a *smooth* approximation of the LP relaxation (2.42) of the original labeling problem (4.2), with the local polytope constraints (2.41) *built in*.

Remark 4.1 (Role of smoothing) The influence of the smoothing parameter τ will be examined in detail in the remainder of this chapter. However, we wish to point out from the beginning that the ability of our smooth geometric approach to compute *integral* labelings does *not* necessarily imply values of $\tau\approx 0$ close to zero, because the rounding mechanism to integral assignments is a *different one*, as will be shown in Section 4.4. As a consequence, larger feasible values of τ weaken the nonlinear relation (4.6) and considerably speed up the convergence of numerical algorithm for iterative label assignment. △

Remark 4.2 (Validity of the local polytope constraints) Using the regularized local Wasserstein distances (4.6) implies that the local marginalization constraints (2.41) are *always* satisfied. This is in sharp contrast to *loopy belief propagation*, were these constraints are gradually enforced during the iteration and are guaranteed to hold only *after* convergence of the entire iteration process. We discuss this fact in Section 4.4.2 in more detail. △

In order to get an intuition about suitable smoothing functions F_τ, we inspect the smoothed local Wasserstein distance (4.6) in more detail. To this end, we pick out and fix any pair of vertices $i,j\in\mathcal{V}$ connected by an edge $ij\in\mathcal{E}$ and simplify our notation in the remainder of this section by dropping indices as follows.

$$M = \mu_{ij}\in\mathbb{R}^{n\times n}, \qquad \text{(coupling measure)} \tag{4.8a}$$

$$\Theta = \theta_{ij}\in\mathbb{R}^{n\times n}, \qquad \text{(transportation costs)} \tag{4.8b}$$

$$\mu = \begin{pmatrix}\mu_i\\ \mu_j\end{pmatrix} = \begin{pmatrix}M\mathbb{1}_n\\ M^\top\mathbb{1}_n\end{pmatrix}, \qquad \text{(stacked marginals)} \tag{4.8c}$$

$$\nu = \begin{pmatrix} \nu_i \\ \nu_j \end{pmatrix}, \qquad \text{(stacked dual vectors)} \tag{4.8d}$$

In this notation, the local (non-smooth) Wasserstein distance (4.5b) reads

$$d_\Theta(\mu_i, \mu_j) = \min_{M \in \Pi(\mu_i, \mu_j)} \langle \Theta, M \rangle, \tag{4.9}$$

for any edge $ij \in \mathcal{E}$. Using the linear map $\mathcal{A}$ defined by (1.11a), we rewrite expression (4.9) as

$$d_\Theta(\mu_i, \mu_j) = \min_{M \in \mathbb{R}^{n \times n}} \langle \Theta, M \rangle \quad \text{s.t.} \quad \mathcal{A}M = \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix}, \quad M \geq 0\,. \tag{4.10}$$

The corresponding dual LP of (4.10) is given by

$$\max_{\nu \in \mathbb{R}^{2n}} \langle \mu, \nu \rangle \quad \text{s.t.} \quad \mathcal{A}^\top \nu \leq \Theta\,. \tag{4.11}$$

So we do the same for the *smoothed* local Wasserstein distance (4.6) which is given by

$$\begin{aligned} d_{\Theta,\tau}(\mu_i, \mu_j) &:= \min_{M \in \mathbb{R}^{n \times n}} \langle \Theta, M \rangle + F_\tau(M) \quad \text{s.t.} \quad \mathcal{A}M = \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix}, \quad M \geq 0, \\ &= \min_{M \in \mathbb{R}^{n \times n}} \langle \Theta, M \rangle + F_\tau(M) + \delta_{\mathbb{R}^{n \times n}_+}(M) + \delta_{\{0\}}\left(\mathcal{A}M - \begin{pmatrix} \mu_i \\ \mu_j \end{pmatrix}\right), \end{aligned} \tag{4.12}$$

for $F_\tau \in \mathcal{F}_0$ and $\tau > 0$, and the corresponding dual problem is given by

$$\max_{\nu \in \mathbb{R}^{2n}} \langle \mu, \nu \rangle - G_\tau^*\left(\mathcal{A}^\top \nu - \Theta\right), \tag{4.13}$$

where G_τ^* is the conjugate function of

$$G_\tau(M) = F_\tau(M) + \delta_{\mathbb{R}^{n \times n}_+}(M). \tag{4.14}$$

Suitable candidates of functions G_τ for smoothing d_Θ suggest themselves by comparing the dual LPs (4.11) and (4.13). Rewriting the constraints of (4.11) in the form

$$\delta_{\mathbb{R}^{n \times n}_-}(\mathcal{A}^\top \nu - \Theta) \tag{4.15}$$

and comparing with (4.13) shows that G_τ^* should be a smooth approximation of the indicator function $\delta_{\mathbb{R}^{n \times n}_-}$. We get back to this point in Section 4.3.2.

4.2 Global Euclidean Gradient

In this section we compute the Euclidean gradient ∇E_τ of the objective function (4.7). In general, if the pairwise model parameters $\theta_\mathcal{E}$ are not symmetric, then the smoothed local Wasserstein distances are not symmetric either:

$$\theta_{ij} \neq \theta_{ij}^\top \implies d_{\theta_{ij},\tau}(\mu_i,\mu_j) \neq d_{\theta_{ij},\tau}(\mu_j,\mu_i), \quad ij \in \mathcal{E}. \tag{4.16}$$

Therefore, we introduce an *arbitrary fixed orientation* (i,j) (ordered pair) of all edges $ij \in \mathcal{E}$, which means $ij \in \mathcal{E} \implies ji \notin \mathcal{E}$. As a consequence, (4.7) reads

$$E_\tau(\mu_\mathcal{V}) = \sum_{i\in\mathcal{V}} \Big(\langle \theta_i, \mu_i \rangle + \sum_{j\colon (i,j)\in\mathcal{E}} d_{\theta_{ij},\tau}(\mu_i,\mu_j) \Big). \tag{4.17}$$

The following proposition specifies the gradient ∇E_τ in terms of the local gradients of the smoothed Wasserstein distances $d_{\theta_{ij},\tau}$. These latter gradients are studied in Section 4.3.1 (Theorem 4.5).

Proposition 4.2 (Euclidean gradient of (4.7))
Suppose the edges $\mathcal{E}$ have an arbitrary fixed orientation. Then, the i-th row of the Euclidean gradient $\nabla E_\tau(\mu_\mathcal{V}) \in T^m$ at $\mu_\mathcal{V} \in \mathcal{W}$ of the objective function (4.7) is given by

$$\nabla_i E_\tau(\mu_\mathcal{V}) = \Pi_T(\theta_i) + \sum_{j\colon (i,j)\in\mathcal{E}} \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j) + \sum_{j\colon (j,i)\in\mathcal{E}} \nabla_2 d_{\theta_{ji},\tau}(\mu_j,\mu_i)\,, \tag{4.18}$$

where $\nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j) \in T$ and $\nabla_2 d_{\theta_{ji},\tau}(\mu_j,\mu_i) \in T$ are the Euclidean gradients of the smoothed Wasserstein distances

$$d_{\theta_{ij},\tau}(\cdot,\mu_j)\colon \mathcal{S} \to \mathbb{R}, \qquad \text{and} \qquad d_{\theta_{ji},\tau}(\mu_j,\cdot)\colon \mathcal{S} \to \mathbb{R}. \tag{4.19}$$

Proof See Appendix A.3. □

Now, we consider the specific case that all pairwise model parameters are *symmetric*, i.e. $\theta_{ij} = \theta_{ij}^\top$ (cf. Corollary 4.4). Due to definition (2.41b), the set $\Pi(\cdot,\cdot)$ of coupling measures has marginals as arguments. We start with the following preparatory Lemma.

Lemma 4.3
Suppose the convex smoothing function F_τ, which defines the regularized local Wasserstein distances (4.6), satisfies $F_\tau(M) = F_\tau(M^\top)$ for all $M \in \Pi(W_i,W_j)$. Then,

the Wasserstein distance satisfies

$$d_{\theta_{ij},\tau}(\mu_i,\mu_j) = d_{\theta_{ij}^\top,\tau}(\mu_j,\mu_i). \tag{4.20}$$

△

Proof See Appendix A.3. □

As a consequence of Lemma 4.3, if all pairwise model parameters θ_{ij} are symmetric and the assumption $F_\tau(M) = F_\tau(M^\top)$ holds for all $M \in [0,1]^{n\times n}$, then we do not need to choose an edge orientation as was done in connection with (4.17). By using the set $\mathcal{N}(i)$, which denotes all neighbor vertices of vertex i (1.10), we can rewrite (4.17) as

$$E_\tau(\mu_\mathcal{V}) = \sum_{i\in\mathcal{V}} \Big(\langle \theta_i, \mu_i \rangle + \frac{1}{2} \sum_{j\in\mathcal{N}(i)} d_{\theta_{ij},\tau}(\mu_i,\mu_j) \Big). \tag{4.21}$$

The following corollary reformulates Proposition 4.2 accordingly.

Corollary 4.4 (Euclidean gradient of (4.7): Symmetric case)
Suppose $F_\tau(M) = F_\tau(M^\top)$ for all $M \in [0,1]^{n\times n}$ and θ_{ij} is symmetric for all $ij \in \mathcal{E}$. Then, the i-th row of the Euclidean gradient ∇E_τ is given by

$$\nabla_i E_\tau(\mu_\mathcal{V}) = \Pi_T(\theta_i) + \sum_{j\in\mathcal{N}(i)} \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j). \tag{4.22}$$

Proof See Appendix A.3. □

4.3 Local Wasserstein Gradient

4.3.1 Formula of the Local Wasserstein Gradient

In this section we check the differentiability of the smoothed Wasserstein distance $d_{\theta_{ij},\tau}(\mu_i,\mu_j)$, $ij \in \mathcal{E}$, and specify an expression for the corresponding gradient. The following theorem formulates the main results of this section. Note, that we again use the simplified notation (4.8).

Theorem 4.5 (Euclidean gradient of Wasserstein distance)
Consider $\mathcal{S} \subset \mathbb{R}^n$ as an Euclidean submanifold with tangent space T defined by (3.2), and let

$$g(\mu,\nu) = \langle \mu, \nu \rangle - G_\tau^*(\mathcal{A}^\top \nu - \Theta) \tag{4.23}$$

denote the dual objective function (4.27). Then the smoothed Wasserstein distance $d_{\Theta,\tau} : \mathcal{S} \times \mathcal{S} \to \mathbb{R}$ is differentiable, and the Euclidean gradient of $d_{\Theta,\tau}$ at $p = (p_1, p_2) \in \mathcal{S} \times \mathcal{S}$ is given by

$$\nabla d_{\Theta,\tau}(p) = \nabla d_{\Theta,\tau}(p_1, p_2) = \overline{\nu}_T := \Pi_{T\times T}(\overline{\nu}) = \begin{pmatrix} \Pi_T(\overline{\nu}_1) \\ \Pi_T(\overline{\nu}_2) \end{pmatrix}, \tag{4.24}$$

where

$$\overline{\nu} = \begin{pmatrix} \overline{\nu}_1 \\ \overline{\nu}_2 \end{pmatrix} \in \operatorname*{argmax}_{\nu\in\mathbb{R}^{2n}} g(p, \nu). \tag{4.25}$$

Proof See Appendix A.3. □

The basic idea of the proof of Theorem 4.5 (see Appendix A.3) is to apply Theorem 1.2. In order to do so, we have to check if the premises of Theorem 1.2 hold in our situation. We start with some preparatory lemmas, that also clarify the structure of the dual solution set. In particular, this set restricted to $\operatorname{im}(\mathcal{A})$ is a singleton, i.e. the set consists of exactly one element (Lemma 4.10). The overall line of reasoning continues on in Section 4.3.2.

Lemma 4.6 (Dual problem of Wasserstein distance)
Let $G_\tau(M) = F_\tau(M) + \delta_{\mathbb{R}_+^{n\times n}}(M)$ with the convex smoothing function F_τ of Equation (4.6), and assume the conjugate function G_τ^* is continuously differentiable. Then the dual problem of

$$\min_{M\in\Pi(\mu_i,\mu_j)} \left\{\langle \Theta, M\rangle + F_\tau(M)\right\} \tag{4.26}$$

is given by

$$\max_{\nu_1,\nu_2} \left\{\langle \mu, \nu\rangle - G_\tau^*(\mathcal{A}^\top \nu - \Theta)\right\}. \tag{4.27}$$

Furthermore, assuming that strong duality holds, the conditions for optimal primal $\overline{M}$ and dual $\overline{\nu} = (\overline{\nu}_1, \overline{\nu}_2)$ solutions are

$$\overline{M} = \nabla G_\tau^*\left(\mathcal{A}^\top \overline{\nu} - \Theta\right), \qquad \mathcal{A}^\top \overline{\nu} - \Theta \in \partial G_\tau(\overline{M}) \tag{4.28a}$$

together with the affine constraint

$$\mathcal{A}\overline{M} = \mu. \tag{4.28b}$$

Proof See Appendix A.3. □

Remark 4.3 (Smoothness of G_τ^*) The *smoothness* assumption with respect to G_τ^* enables to compute conveniently the gradient of the smoothed Wasserstein distance $d_{\Theta,\tau}$. It corresponds to a *convexity* assumption on G_τ. These aspects are further discussed in Section 4.3.2 as well. △

Remark 4.4 (Strong duality) The condition of strong duality (cf. [13, Section I.5]) made by Lemma 4.6 is crucial for what follows. This condition will be satisfied later on when working in a *geometric* setting with local measures M, μ_i, μ_j with *full* support, as introduced in Section 3.1.2. △

The following Lemma characterizes the kernel of the linear mapping $\mathcal{A}^\top$, defined by (1.11b).

Lemma 4.7 (Kernel of linear map (1.11b))
Let the linear mapping $\mathcal{A}^\top$ be defined by (1.11b). Then, its kernel is given by

$$\ker(\mathcal{A}^\top) = \left\{ \lambda \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \in \mathbb{R}^{2n} \colon \lambda \in \mathbb{R} \right\} \tag{4.29a}$$

and its orthogonal complement by

$$\ker(\mathcal{A}^\top)^\perp = \left\{ x \in \mathbb{R}^{2n} \colon \left\langle x, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle = 0 \right\}. \tag{4.29b}$$

Proof See Appendix A.3. □

The following Lemma characterizes the set of optimal dual solutions of problem (4.27).

Lemma 4.8 (Set of optimal dual solutions)
Let the function G_τ^* of the dual objective function (4.27) resp. (4.23) be continuously differentiable and strictly convex, and let $p \in \mathbb{R}^{2n}_{++}$. Then the set of optimal dual solutions has the form

$$\operatorname*{argmax}_{v \in \mathbb{R}^{2n}} g(p, v) = \begin{cases} \{\overline{v}\}, & \text{if } \left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle \neq 0, \quad (4.30a) \\ \overline{v} + \ker(\mathcal{A}^\top), & \text{if } \left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle = 0. \quad (4.30b) \end{cases}$$

Proof See Appendix A.3. □

In view of Theorem 1.2, the set of maximizers has to be a single element. As characterized by the previous Lemma 4.8, this is only the case if $\left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle \neq 0$. In our situation, i.e. the case $\left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle = 0$, we still have work to do. We ensure uniqueness of the dual maximizer by further restricting the set of optimal dual solutions which is clarified by Lemma 4.10.

Lemma 4.9 (Orthogonal decomposition of $\mathbb{R}^{2n}$)
Let the linear mappings $\mathcal{A}$ and $\mathcal{A}^\top$ be defined by (1.11a) and (1.11b), respectively. Then, the following orthogonal decomposition

$$\mathbb{R}^{2n} = \ker(\mathcal{A}^\top) \oplus \operatorname{im}(\mathcal{A}) \tag{4.31}$$

into linear subspaces applies. We denote the corresponding components of a vector $v \in \mathbb{R}^{2n}$ by $v = v_{\ker} + v_{\operatorname{im}}$.

Proof See Appendix A.3. □

Lemma 4.10 (Restricted set of optimal dual solutions)
Consider the orthogonal decomposition (4.31) and denote the corresponding components of a vector $v \in \mathbb{R}^{2n}$ by $v = v_{\ker} + v_{\operatorname{im}}$. Then, for $p \in \mathbb{R}^{2n}_{++}$ satisfying $\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \rangle = 0$, we have

$$\operatorname*{argmax}_{v_{\operatorname{im}} \in \operatorname{im}(\mathcal{A})} g(p, v_{\operatorname{im}}) = \{\overline{v}_{\operatorname{im}}\}, \tag{4.32a}$$

where $\overline{v}_{\operatorname{im}} = \Pi_{\operatorname{im}(\mathcal{A})}(\overline{v})$ for any $\overline{v} \in \operatorname{argmax}_{v \in \mathbb{R}^{2n}} g(p, v)$. Furthermore, the corresponding objective value satisfies

$$g(p, \overline{v}_{\operatorname{im}}) = \max_{v_{\operatorname{im}} \in \operatorname{im}(\mathcal{A})} g(p, v_{\operatorname{im}}) = \max_{v \in \mathbb{R}^{2n}} g(p, v). \tag{4.32b}$$

In other words, a dual maximizer $\overline{v}_{\operatorname{im}}$ exists and is unique in the subspace $\operatorname{im}(\mathcal{A})$.

Proof See Appendix A.3. □

4.3.2 Computation of the Local Wasserstein Gradient

A core subroutine of our approach concerns the computation of the local Wasserstein gradients as part of the overall gradient (4.18). In this section we show why

the *negative entropy function*, that we use in our implementation for smoothing the local Wasserstein distances, plays a distinguished role.

Using again the simplified notation (4.8), the smooth entropy regularized Wasserstein distance (4.6) reads

$$d_{\Theta,\tau}(\mu_i,\mu_j) = \min_{M\in\mathbb{R}^{n\times n}} \langle \Theta, M\rangle - \tau H(M) \quad \text{s.t.} \quad \mathcal{A}M = \begin{pmatrix}\mu_i\\ \mu_j\end{pmatrix}, \quad M \geq 0\,, \tag{4.33a}$$

with the entropy function

$$H(M) = -\sum_{i,j} M_{i,j} \log M_{i,j}. \tag{4.33b}$$

As derived in the previous section and formulated in Theorem 4.5, the gradients of (4.33) are the maximizer of the corresponding dual problem. This dual problem is given by

$$\max_{\nu\in\mathbb{R}^{2n}} \langle \mu, \nu\rangle - \tau \sum_{k,l} \exp\Big[\frac{1}{\tau}\Big(\mathcal{A}^\top \nu - \Theta\Big)_{k,l}\Big]. \tag{4.34}$$

In view of the general form (4.13) of this dual problem, the indicator function (4.15) is smoothly approximated by $\tau\exp(\frac{1}{\tau}x)$. Figure 4.1 compares this approximation and the classical logarithmic barrier $-\log(-x)$ function for approximating the indicator function $\delta_{\mathbb{R}_-}$ of the non-positive orthant. Log-barrier penalty functions are the method of choice for *interior point methods* [49, 72], which *strictly* rule out violations of the constraints. While this is essential for many applications where constraints represent physical properties that cannot be violated, it is *not* essential in the present case for calculating the Wasserstein messages. Moreover, the bias towards interior points by log-barrier functions, as Figure 4.1 clearly shows, is detrimental in the present context and favours the formulation (4.34).

We now derive how the local Wasserstein gradients (4.24) are computed based on the formulation (4.33) and examine numerical aspects depending on the smoothing parameter τ. It is well known that doubly stochastic matrices as solutions of convex programs like (4.33) can be computed by iterative matrix scaling [70, 68], [16, Ch. 9]. This has been made popular in the field of machine learning by [20].

For the *entropy regularization*, the optimality condition (4.28) takes the form

$$\overline{M} = \exp\Big[\frac{1}{\tau}\Big(\mathcal{A}^\top \overline{\nu} - \Theta\Big)\Big], \tag{4.35}$$

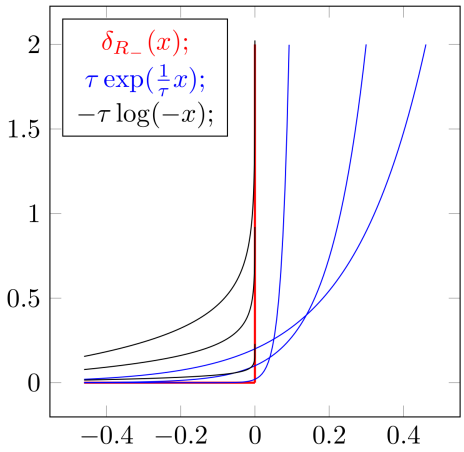


Figure 4.1: Approximations of the indicator function $\delta_{\mathbb{R}_-}$ of the non-positive orthant. The log-barrier function (black curves) strictly rules out violations of the constraints but induce a bias towards interior points. Our formulation (blue curves) is less biased and reasonable approximates the δ-function (red curve) depending on the smoothing parameter τ. Displayed are the approximations of $\delta_{\mathbb{R}_-}$ for $\tau = \frac{1}{5}, \frac{1}{10}, \frac{1}{50}$.

and rearranging yields the connection to matrix scaling:

$$\begin{aligned} \overline{M} = \exp\Big[\frac{1}{\tau}\Big(\mathcal{A}^\top \overline{v} - \Theta\Big)\Big] &\overset{(1.11b)}{=} \exp\Big[\frac{1}{\tau}\Big(\overline{v}_1 \mathbb{1}_n^\top + \mathbb{1}_n \overline{v}_2^\top - \Theta\Big)\Big] \\ &= \big(\exp(\tfrac{\overline{v}_i}{\tau})\exp(\tfrac{\overline{v}_j}{\tau})^T\big) \cdot \exp\big(-\tfrac{1}{\tau}\Theta\big) \\ &= \mathrm{Diag}\big(\exp(\tfrac{\overline{v}_i}{\tau})\big)\exp\big(-\tfrac{1}{\tau}\Theta\big)\mathrm{Diag}\big(\exp(\tfrac{\overline{v}_j}{\tau})\big), \end{aligned} \tag{4.36}$$

where Diag$(\cdot)$ denotes the diagonal matrix with the argument vector as entries. For given marginals $\mu = (\mu_i, \mu_j)$ due to (4.33) and with the shorthand $K = \exp\big(-\frac{1}{\tau}\Theta\big)$, the optimal dual variables $\overline{v} = (\overline{v}_1, \overline{v}_2)$ can be determined by the Sinkhorn's iterative algorithm [70], up to a common multiplicative constant. Specifically, we have

Lemma 4.11 ([20, Lemma 2])
For $\tau > 0$ and $K = \exp\big(-\frac{1}{\tau}\Theta\big)$, the solution $\overline{M}$ of (4.33) is unique and has the form $\overline{M} = \mathrm{diag}(v_i) K \mathrm{diag}(v_j)$, where the two vectors $v_i, v_j \in \mathbb{R}^n$ are uniquely defined up to a multiplicative factor.

Proof See [20, Lemma 2]. □

Accordingly, by setting

$$v_i := \exp(\tfrac{v_1}{\tau}), \qquad v_j := \exp(\tfrac{v_2}{\tau}), \tag{4.37}$$

Sinkhorn's fixed point iterations [70] read

$$v_i^{(k+1)} = \frac{\mu_i}{K\left(\frac{\mu_j}{K^\top v_i^{(k)}}\right)}, \qquad v_j^{(k+1)} = \frac{\mu_j}{K^\top\left(\frac{\mu_i}{K v_j^{(k)}}\right)}, \tag{4.38}$$

which are iterated until the change between consecutive iterates is small enough. Denoting the iterates after convergence by $\overline{v}_i, \overline{v}_j$, resubstitution into (4.37) determines the optimal dual variables

$$\overline{v}_i = \tau \log \overline{v}_i, \qquad \overline{v}_j = \tau \log \overline{v}_j. \tag{4.39}$$

Due to Theorem 4.5, the local Wasserstein gradients then finally are given by

$$\nabla d_{\Theta,\tau}(\mu_i, \mu_j) = \begin{pmatrix} \Pi_T(\overline{v}_i) \\ \Pi_T(\overline{v}_j) \end{pmatrix}, \tag{4.40}$$

where the projection Π_T due to (3.3) removes the common multiplicative constant resulting from Sinkhorn's algorithm.

While the linear convergence rate of Sinkhorn's algorithm is known theoretically [41], the numbers of iterations required in practice significantly depends on the smoothing parameter τ. In addition, for smaller values of τ, an entry of the matrix $K = \exp\left(-\frac{1}{\tau}\Theta\right)$ might be too small to be represented on a computer, due to machine precision. As a consequence, the matrix K might have entries which are numerically treated as zeros and Sinkhorn's algorithm does not necessarily converge to the true optimal solution.

Fortunately, our approach does allow larger values of τ because merely a sufficiently accurate approximation of the *gradient* of the Wasserstein distance is required, rather than an approximation of the Wasserstein distance itself, to obtain valid *descent* directions. Figures 4.2 and 4.3 demonstrate that this indeed holds for relatively large values of τ, e.g. $\tau \in \{\frac{1}{5}, \frac{1}{10}, \frac{1}{15}\}$, no matter if the number of labels is $n = 10$ or $n = 1000$.

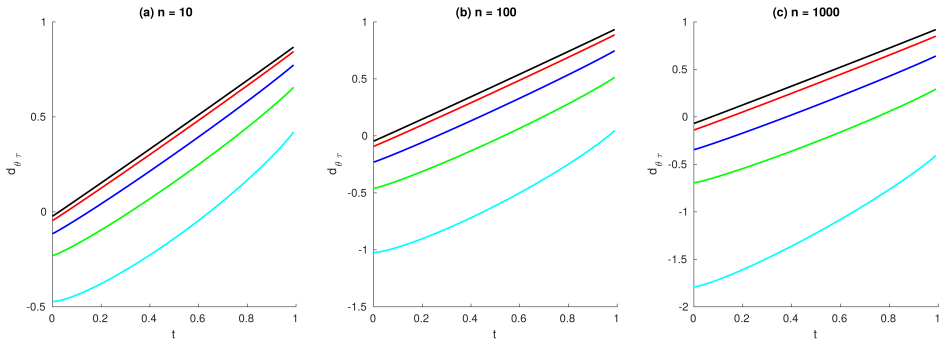


Figure 4.2: Entropy-regularized Wasserstein distance $d_{\Theta,\tau}(c,\gamma(t))$ for varying parameter τ and increasing numbers n of labels. The plots show the entropy-regularized Wasserstein distance $d_{\Theta,\tau}(c,\gamma(t))$ for varying parameter τ and increasing numbers n of labels. The line segment $\gamma(t) = t(e_1 - c) + c \in \Delta_n$, with $t \in [0,1]$, connects the barycenter $c = \frac{1}{n}\mathbb{1}$ and the vertex e_1 on the simplex Δ_n. The cost matrix Θ is given by the Potts prior (4.4). In all three plots the parameter τ is set to $\tau = \frac{1}{5}$ (cyan), $\tau = \frac{1}{10}$ (green), $\tau = \frac{1}{20}$ (blue), $\tau = \frac{1}{50}$ (red) and $\tau = \frac{1}{100}$ (black). Even though the values of the approximation of the distance itself differ considerably, the *slope* of the distance is already approximated quite well for larger values of τ (uniformly for both small and large numbers n of labels).

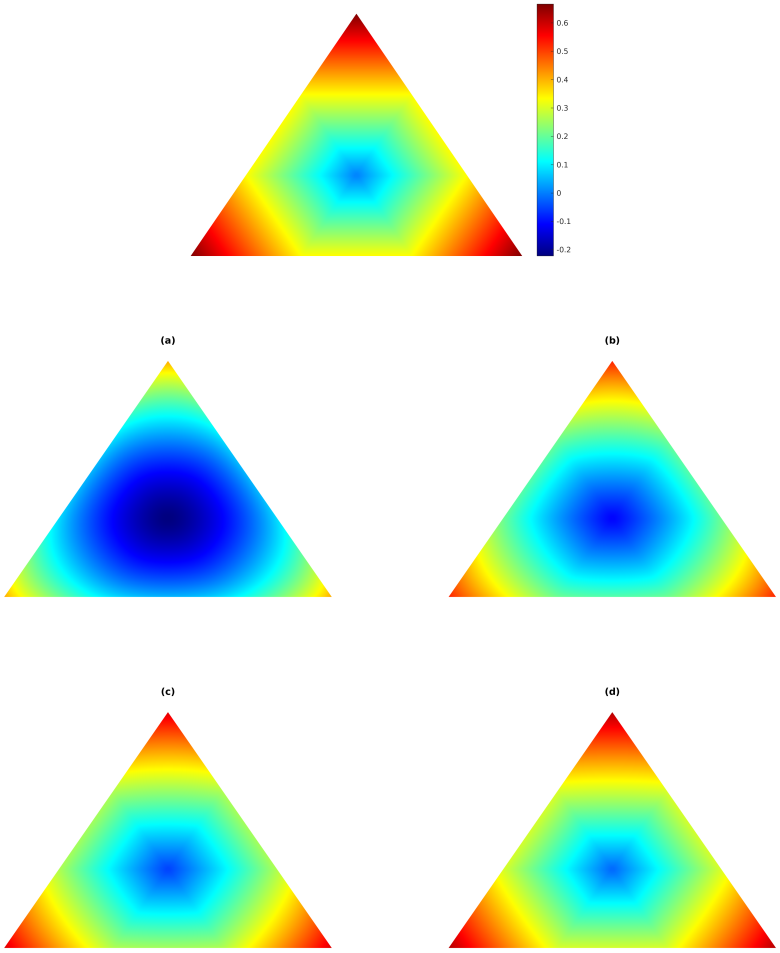


Figure 4.3: Wasserstein distance with Potts prior (4.4) on probability simplex Δ_3. The plot shows the exact Wasserstein distance (top) compared to the entropy-regularized Wasserstein distance with the Potts prior (4.4) from the barycenter to every point on Δ_3 for different values of τ: (a) $\tau = \frac{1}{5}$, (b) $\tau = \frac{1}{10}$, (c) $\tau = \frac{1}{20}$ and (d) $\tau = \frac{1}{50}$. These plots confirm that even for relatively large values of τ, e.g. $\frac{1}{10}$ and $\frac{1}{20}$, the *gradient* of the Wasserstein distance is sufficiently accurate approximated so as to obtain valid descent directions for distance minimization.

4.4 Graphical Models on the Assignment Manifold

In this section we explain how image labeling, based on a given graphical model, can be performed on the assignment manifold (3.18) using the global and local gradients derived in sections 4.2 and 4.3, respectively. The graphical model is given in terms of an energy function $E(x)$ of the form (4.2). The basic idea for determining a labeling x with low energy $E(x)$ is to combine minimization of the convex relaxation (2.42) and non-convex rounding to an integral solution in a *single smooth process*. This idea is worked out in Section 4.4.1 and is realized by restricting the smooth approximation (4.7) of the objective function to the assignment manifold. The numerical integration of the corresponding Riemannian gradient flow is combined with a rounding mechanism to integral solutions. In order to highlight the essential properties of our approach as a novel way of *belief propagation* using dually computed local Wasserstein gradients, we complement in Section 4.4.2 our preliminary observations, stated as Remarks 4.1 and 4.2.

4.4.1 Combination of Minimizing and Rounding

We recall how regularization is performed by the assignment approach of [8]: distance vectors (3.40) representing the data term of classical variational approaches are lifted to the assignment manifold by (3.41) and geometrically averaged over spatial neighborhoods (3.42).

Given a graphical model in terms of an energy function (4.2), regularization is already *defined* by the pairwise model parameters $\theta_{ij}(\ell_k, \ell_r)$, so that evaluating the gradient of the regularized objective function (4.7) *implies* averaging over spatial neighborhoods, as (4.18) clearly displays. Minimizing the corresponding Riemannian gradient flow on the assignment manifold with the explicit Euler method leads to the update of the assignment matrix

$$W_i^{(k+1)} = \frac{W_i^{(k)} \cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle W_i^{(k)}, e^{-h\nabla_i E_\tau(W^{(k)})}\rangle}, \quad i \in [m], \qquad h > 0, \qquad W^{(0)} = \frac{1}{n}\mathbb{1}_m\mathbb{1}_n^\top, \tag{4.41}$$

where h is a step-size parameter and the partial gradients $\nabla_i E_\tau(W^{(k)})$ are given by (4.18). The sequence $(W^{(k)})$ is initialized in an unbiased way at the barycenter $W^{(0)} \in \mathcal{W}$.

Remark 4.5 (Notation: W vs. $\mu_\mathcal{V}$) The assignment matrix $W \in \mathcal{W}$ plays the role of the node variables $\mu_\mathcal{V}$ of the basic LP relaxation as defined by (2.38), with relaxed

domain due to (2.41). Unlike μ_i, however, vectors $W_i \in \mathbb{R}^n_{++}$ always have full support and live on the manifold $\mathcal{S}$. △

This update step minimizes the function E_τ, given by (4.7), on the assignment manifold $\mathcal{W}$. In order to converge to an integral solution, i.e. a valid labeling, we consider an extended objective function

$$f_{\tau,\alpha}(W) := E_\tau(W) + \alpha_h H(W), \qquad \alpha_h = \frac{\alpha}{h}, \tag{4.42}$$

where E_τ is given by (4.7) and $H(W)$ denotes the entropy of the assignment matrix W by

$$H(W) = -\langle W, \log W \rangle. \tag{4.43}$$

The following proposition shows that numerically integrating the Riemannian gradient flow of the extended objective function (4.42) results in a flexible multiplicative update combining minimization and rounding.

Proposition 4.12
Numerically integrating the *Riemannian gradient descent flow* of $f_{\tau,\alpha}$, given by (4.42), by geometric Euler steps results in

$$W_i^{(k+1)} = \frac{(W_i^{(k)})^{1+\alpha} \cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle (W_i^{(k)})^{1+\alpha}, e^{-h\nabla_i E_\tau(W^{(k)})} \rangle}, \quad \alpha \geq 0. \tag{4.44}$$

where a rounding mechanism is incorporated by a *rounding parameter* α.

Proof See Appendix A.3. □

Remark 4.6 (Continuous DC programming) Equation (4.42) admits to interpret the update rule (4.44) as a *continuous difference of convex (DC) programming* strategy. Unlike the established DC approach [53, 54], however, which takes *large steps* by solving to optimality a sequence of convex programs in connection with updating an affine upper bound of the concave part of the objective function, our update rule (4.44) differs in two essential ways: *geometric optimization* by numerically integrating the Riemannian gradient flow *tightly interleaves with rounding* to an integral solution. The rounding effect is achieved by minimizing the entropy term of (4.42) which steadily sparsifies the assignment vectors comprising W. △

4.4.2 Connection to Belief Propagation

In this section we discuss the connection of our approach to *belief propagation (BP)*, as already stated in Remarks 4.1 and 4.2. A derivation of BP can be found in Section 2.2.5 and a more detailed version in Appendix B.

From the viewpoint of BP, our alternative approach (4.42) emerges as follows, starting at the smoothed primal LP (2.46) and following the idea of the proof from Lemma 4.1.

$$\min_{\mu\in\mathcal{L}_\mathcal{G}} \langle\theta,\mu\rangle - \varepsilon H(\mu) \tag{4.45a}$$

$$= \min_{\mu\in\mathcal{L}_\mathcal{G}} \langle\theta,\mu\rangle - \varepsilon\Big(\sum_{ij\in\mathcal{E}} H(\mu_{ij}) - \sum_{i\in\mathcal{V}} \big(d(i)-1\big)H(\mu_i)\Big) \tag{4.45b}$$

$$= \min_{\mu\in\mathcal{L}_\mathcal{G}} \langle\theta_\mathcal{V},\mu_\mathcal{V}\rangle + \langle\theta_\mathcal{E},\mu_\mathcal{E}\rangle - \varepsilon \sum_{ij\in\mathcal{E}} H(\mu_{ij}) + \varepsilon \sum_{i\in\mathcal{V}} \big(d(i)-1\big)H(\mu_i) \tag{4.45c}$$

$$= \min_{\mu_\mathcal{V}\in\Delta_n^m} E_\varepsilon(\mu_\mathcal{V}) + \varepsilon \sum_{i\in\mathcal{V}} \big(d(i)-1\big)H(\mu_i). \tag{4.45d}$$

Formulation (4.42) results from replacing ε by a smoothing parameter τ which can be set to a value *not very* close to 0 (cf. Remark 4.1), and we absorb the second nonnegative factor weighting the entropy term by a second parameter α. As demonstrated in Section 4.5, this latter parameter enables to control precisely the trade-off between accuracy of labelings in terms of the given objective function E_τ of (4.42), that approximates the original discrete objective function (4.2), and the speed of convergence to an integral (labeling) solution.

Regarding the resulting term E_τ, a key additional step is to use the reformulation (4.7), because all edge-based variables are *locally* "dualized away", as done *globally* with *all* variables when using established belief propagation (cf. (2.50)). In this way, we can work in the primal domain and with graphs having higher connectivity, without suffering from the enormous memory requirements that would arise from merely smoothing the LP and solving (2.46) in the primal domain. Furthermore, the *messages* defined by our approach have a clear interpretation in terms of the smoothed Wasserstein distance between local marginal measures.

We summarize this discussion by contrasting the following *observations* of established belief propagation and our approach. Regarding belief propagation, we have:

1. *Local non-convexity.* The negative $-H(\mu)$ of the so-called *Bethe entropy* function $H(\mu)$ is *non-convex* in general for graphs $\mathcal{G}$ with cycles [74, Section 4.1], due to the negative sign of the second sum of (2.46).

2. *Local rounding at each step.* The max-product algorithm performs *local rounding* at *every* step of the iteration so as to obtain integral solutions, i.e. a *labeling* after convergence. This operation results as limit of a *non-convex* function, due to (1).

3. *Either non-smoothness or strong nonlinearity.* The latter max-operation is inherently non-smooth. Preferring instead a smooth approximation with $0 < \varepsilon \ll 1$ necessitates to choose ε very small so as to ensure rounding. This, however, leads to *strongly nonlinear* functions of the form (1.15) that are difficult to handle numerically.

4. *Invalid constraints.* Local marginalization constraints are only satisfied *after* convergence of the iteration. Intuitively it is plausible that, by only *gradually* enforcing constraints in this way, the iterative process becomes more susceptible to getting stuck in unfavourable stationary points, due to the non-convexity according to (1).

In contrast, our *geometric approach* defines *message passing* with respect to vertex $i \in \mathcal{V}$ by evaluating the local Wasserstein gradients of (4.18) for all edges incident to i. We therefore call these local gradients *Wasserstein messages* which are passed along edges. Similarly to (2.50), each such message is given by *dual* variables through (4.24), that solve the regularized *local* dual LPs (4.23). As a consequence, local marginalization constraints are *always* satisfied, throughout the iterative process.

In addition, we make the following *observations* in correspondence to the points (1)-(4) above:

1. *Local convexity.* Wasserstein messages of (4.18) are defined by local *convex* programs (4.23). This contrasts with loopy belief propagation and holds true for any pairwise model parameters θ_{ij} of the prior of the graphical model and the corresponding coupling of μ_i and μ_j.

2. *Smooth global rounding after convergence.* Rounding to integral solutions is *gradually* enforced through the Riemannian flow induced by the extended objective function (4.42). In particular, repeated *aggressive* local max operations of the max-product algorithm are replaced by a *smooth* flow.

3. *Smoothness and weak nonlinearity.* The role of the smoothing parameter τ of (4.7) *differs* from the role of the smoothing parameter ε of (2.46). While the latter has to be chosen quite close to 0 so as to achieve rounding at all, τ merely

mollifies the dual local problems (4.23) and hence should be chosen small, but may be considerably larger than ε. In particular, this does not impair rounding due to (2), which happens due to the *global* flow which is *smoothly* driven by the Wasserstein messages. This *decoupling* of smoothing and rounding enables to numerically compute labelings more efficiently. The results of Section 4.5 demonstrate this fact.

4. *Valid constraints.* By construction, computation of the Wasserstein messages enforces all local marginalization constraints *throughout* the iteration. This is in sharp contrast to belief propagation where this generally holds after convergence only. Intuitively, it is plausible that our *more tightly* constrained iterative process is less susceptible to getting stuck in poor local minima. The results of Section 4.5.2 provide evidence of this conjecture.

4.5 Experiments

In this section we demonstrate and discuss the results of our approach using four types of experiments:

1. *Parameter influence.* The dependency of the smoothing parameter τ and the rounding parameter α on the assignment is illustrated (Section 4.5.1).

2. *Cyclic graphical models on $\mathcal{K}^3$.* We comprehensively explore the space of binary graphical models defined on the minimal cyclic graph, the complete graph with three vertices $\mathcal{K}^3$, whose LP relaxation is known to have a substantial part of non-binary vertices. The results exhibit a relationship between α and τ so that in fact a single effective parameter only controls the trade-off between accuracy of optimization and the computational costs (Section 4.5.2).

3. *Comparison to other methods.* A evaluation of our approach together with two established and widely applied approaches, sequential tree-reweighted message passing (TRWS) [42] and loopy belief propagation, reveals similar performance of our approach (Section 4.5.3).

4. *Non-Potts prior.* We demonstrate for a graphical model with *non*-uniform pairwise model parameters (non-Potts prior) that our geometric approach accurately takes them into account (Section 4.5.4).

All experiments are selected to illustrate properties of our approach instead of working out a particular application. In all experiments we use the following setting.

- *Assignment Normalization.* The rounding mechanism addressed by Proposition 4.12 and Remark 4.6 will be effective if α_h in (4.42) is chosen large enough to compensate the influence of the function F_τ that regularizes the local Wasserstein distances (4.6). In this case we avoid numerical problems by the normalization strategy (3.54).

- *Update Schemes.* We use the numerical update scheme (4.44) in our implementation, which reads

$$W_i^{(k+1)} = \frac{(W_i^{(k)})^{1+\alpha} \cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle (W_i^{(k)})^{1+\alpha}, e^{-h\nabla_i E_\tau(W^{(k)})} \rangle}, \quad W_i^{(0)} = \mathbb{1}_{\mathcal{S}_n} = \frac{1}{n}\mathbb{1}_n, \quad i \in \mathcal{V}, k \in \mathbb{N}$$

  where $\alpha \geq 0$ is the *rounding* parameter, $h > 0$ the step-size and τ the *smoothing* parameter for the local Wasserstein distances.

- *Termination Criterion.* As explained in Section 3.4.1, in all experiments we use the normalized averaged entropy as a termination criterion.

4.5.1 Parameter Influence

This experiments illustrates the parameter influence of the *rounding parameter* α and the *smoothing parameter* τ on the assignment. The task is to label a noisy RGB-image $f : \mathcal{V} \to [0,1]^3$, depicted in Fig. 4.4, on the grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with minimal neighborhood size $|\mathcal{N}(i)| = 3 \times 3$, $i \in \mathcal{V}$. As labels we use the prototypical colors $\mathcal{X} = \{\ell_1, \ldots, \ell_8\} \subset [0,1]^3$ (Fig. 4.4, left panel) and as unary potentials

$$\theta_i = \frac{1}{\rho}\big(\|f(i) - \ell_1\|_1, \ldots, \|f(i) - \ell_8\|_1\big), \quad i \in \mathcal{V}, \tag{4.46}$$

with $\|\cdot\|_1$ distance. The scaling factor is set to $\rho = 0.3$. For the pairwise potentials of the model we use a *Potts prior* (4.4) with $\lambda = 1$. Furthermore, we use a constant step-size $h = 0.1$ for numerically integrating the Riemannian descent flow, and we set the threshold for the normalized average entropy termination criterion (3.55) to 10^{-4}.

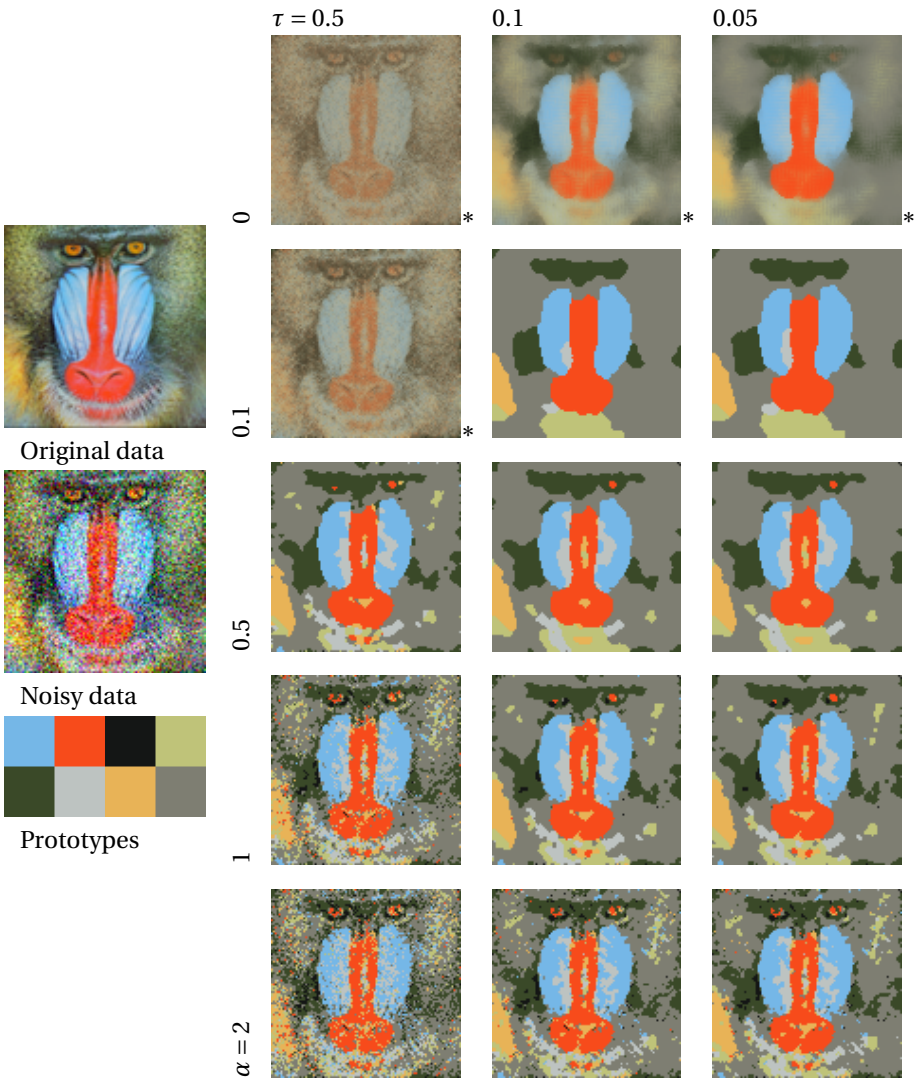


Figure 4.4: Influence of the rounding parameter α and the smoothing parameter τ on the assignment. This plot shows the parameter influence of α and τ on the assignment of 8 prototypical labels to noisy input data. All images marked with an '$*$' do not show integral solutions due to smoothing too strongly the Wasserstein distance in terms of τ relative to α, which overcompensates the effect of rounding. Likewise, smoothing too strongly the Wasserstein distance (left column, $\tau = 0.5$) yields poor approximations of the objective function gradient and to erroneous label assignments. The remaining parameter regime, i.e. smoothing below a reasonably large upper bound $\tau = 0.1$, leads to fast numerical convergence, and the label assignment can be precisely controlled by the rounding parameter α.

Fig. 4.4 shows the influence of the rounding parameter α and the smoothing parameter τ for the Wasserstein distance on the labeling result. All images marked with '$*$' in the lower right corner do not constitute an integral solution, which means that the normalized average entropy (3.55) of the assignment vectors W_i did not drop below the threshold during the iterations. Thus, even though the assignments show a clear tendency, they are not close to an integral solution. This is not a deficiency of our approach but must happen if either no rounding is performed ($\alpha = 0$) or if the influence of rounding is too small compared to the smoothing of the Wasserstein distance (e.g. $\alpha = 0.1$ and $\tau = 0.5$). Increasing the strength of rounding (larger α) leads to a faster decrease in entropy (cf. Fig. 4.5 for the case of $\tau = 0.1$) and therefore to an earlier convergence of the process to a specific labeling. Thus, a more aggressive rounding scheme yields a less regularized result due to the rapid decision for a labeling at an early stage of the algorithm.

The empirical convergence rate depending on the rounding parameter α is displayed by Fig. 4.5, top. A fixed value of the smoothing parameter $\tau = 0.1$ ensures a sufficiently accurate approximation of the Wasserstein distance gradients and hence of the Riemannian descent flow. In addition, the interplay between minimizing the smoothed energy E_τ (4.7) and the rounding mechanism, induced by the entropy H (4.43) in $f_{\tau,\alpha}$ (4.42), is illustrated by Fig. 4.5, bottom. Less aggressive rounding (smaller values of α) leads to a more accurate numerical integration of the flow using a larger number of iterations, and thus to higher quality label assignments with a lower energy of the objective function. We demonstrate this latter aspect quantitatively in Section 4.5.2. For too small values of the rounding parameter α, the algorithm does naturally not converge to an integral solution.

On the other hand, choosing the smoothing parameter τ too large lead to poor approximations of the Wasserstein distance gradients and consequently to erroneous non-regularized labelings, as displayed by Fig. 4.4 (left column) corresponding to $\tau = 0.5$. Once τ is small enough (in our experiments: $\tau < 0.1$) the Wasserstein distance gradients are properly approximated, and the label assignment is regularized and can be controlled by α. In particular, this upper bound on τ is sufficiently large to ensure very rapid convergence of the fixed point iteration for computing the Wasserstein distance gradients.

Fig. 4.6 shows the connection between the objective function $f_{\tau,\alpha}$ (4.42) and the discrete energy E (4.2) of the underlying graphical model. Minimizing $f_{\tau,\alpha}$ (*yellow curve*) using our approach also minimizes the discrete energy E (*violet curve*), which is calculated by rounding the assignment vectors after each iterative step. Furthermore, the interplay between the smoothed energy (4.7) E_τ (*orange curve*) and the entropy H (4.43) (*blue curve*) of $f_{\tau,\alpha} = E_\tau + \alpha H$ is shown. These curves illustrate (i)

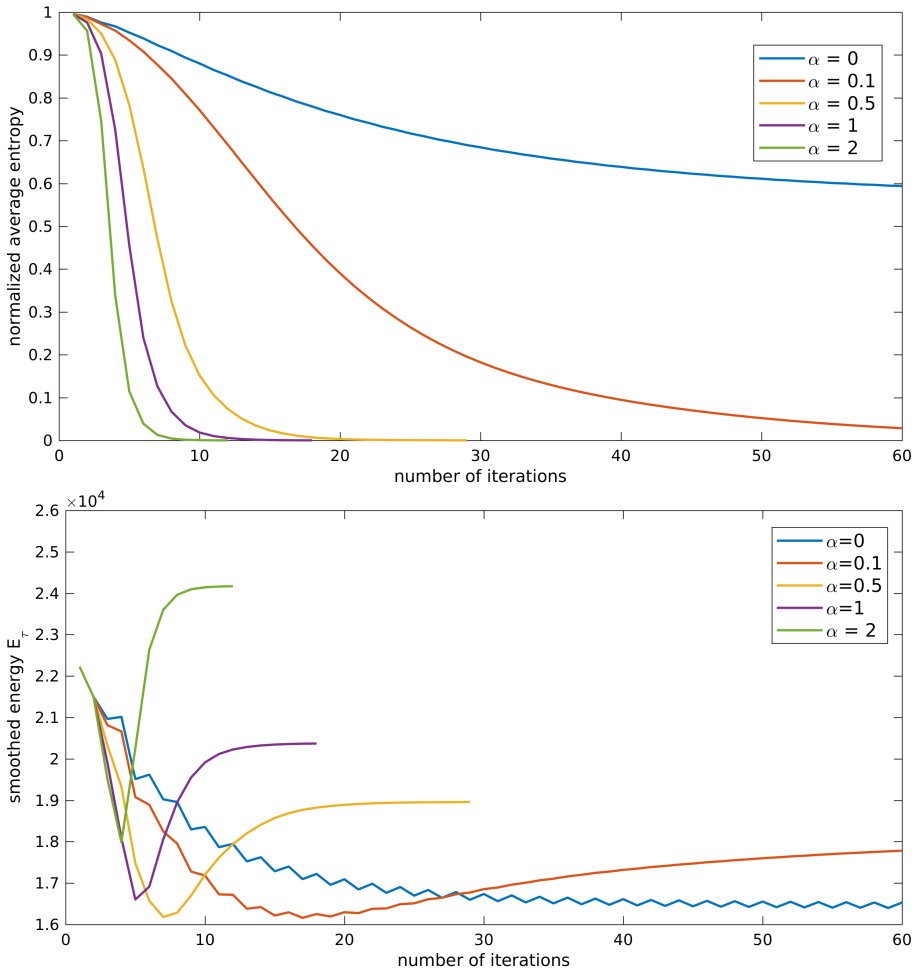


Figure 4.5: Normalized average entropy (3.55) **and smoothed energy** E_τ (4.7)**.** The normalized average entropy (3.55) (TOP) and the smoothed energy E_τ (4.7) (BOTTOM) with smoothing parameter $\tau = 0.1$ are shown. TOP: By increasing the rounding parameter α, the entropy drops more rapidly and hence converges faster to an integral labeling. BOTTOM: Two phases of the algorithm depending on the values for α are clearly visible. In the first phase, the smoothed energy E_τ is minimized up to the point where rounding takes over in the second phase. Accordingly, the sequence of energy values first drops down to lower values corresponding to the problem *relaxation* and then adopts a higher energy level corresponding to an *integral* solution. For smaller α, the algorithm spends more time on minimizing the smoothed energy. This generally results in lower energy values even *after* rounding, i.e. in higher quality labelings.

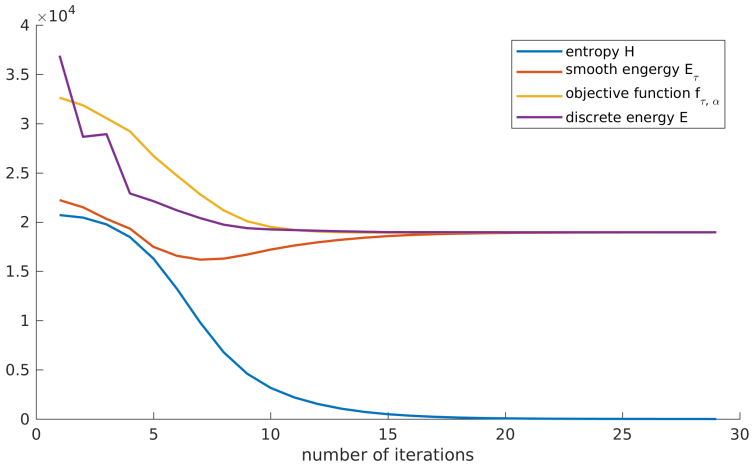


Figure 4.6: Connection between the objective function $f_{\tau,\alpha}$ (4.42) and the discrete energy E (4.2) of the underlying graphical model. For this plot the rounding parameter was fixed $\alpha = 0.5$. Minimizing $f_{\tau,\alpha}$ (*yellow*) by our approach also minimizes E (*violet*), which was calculated for this illustration by rounding the assignment vectors at every iterative step. Additionally, as already discussed in more detail in connection with Fig. 4.5, the interplay between the two terms of $f_{\tau,\alpha} = E_\tau + \alpha H$ is shown, where E_τ (*orange*) denotes the smoothed energy (4.7) and H (*blue*) the entropy (4.43) causing rounding.

the smooth combination of optimization and rounding into a single process, and (ii) that the original discrete energy (4.2) is effectively minimized by this smooth process.

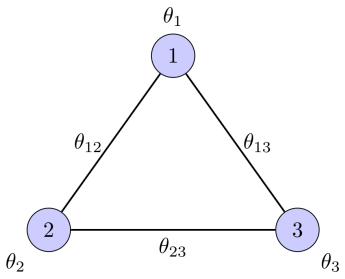
4.5.2 Cyclic Graphical Models on $\mathcal{K}^3$

In this experiment we explore all possible binary models, i.e. $\mathcal{X} = \{0,1\}$, on the minimal cyclic graph $\mathcal{K}^3$ (Fig. 4.7, left panel). Due to the single cycle, models exist where the LP relaxation (2.42) returns a non-binary solution (red part of Fig. 4.7, right panel). As a consequence, evaluating such models with our geometric approach for minimizing (4.7) enables to check two cases:

1. *Binary solution.* Whenever solving the LP relaxation (2.42) by convex programming returns the global binary minimum of (4.2) as solution, we assess if our geometric approach based on the smooth approximation (4.7) returns this solution as well.

2. *Non-binary solution.* Whenever the LP relaxation has a *non-binary* vector as global solution, which therefore is *not* optimal for the labeling problem (4.2), we assess the rounding property of our approach by comparing the result with the *correct* binary labeling globally minimizing (4.2).

The graph $\mathcal{K}^3$ enables us to specify the *marginal polytope* $\mathcal{M}_{\mathcal{K}^3}$ whose vertices are the feasible binary combinatorial solutions that correspond to valid labelings (cf. Section 2.2.4), and to examine the difference to the local polytope $\mathcal{L}_{\mathcal{K}^3}$ whose representation only involves a subset of the constraints corresponding to $\mathcal{M}_{\mathcal{K}^3}$.



μ_1	μ_2	μ_3	μ_{12}	μ_{13}	μ_{23}
0	0	0	0	0	0
0	0	1	0	0	0
0	1	0	0	0	0
0	1	1	0	0	1
1	0	0	0	0	0
1	0	1	0	1	0
1	1	0	1	0	0
1	1	1	1	1	1
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	0	0
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0	$\frac{1}{2}$
$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	$\frac{1}{2}$	0

Figure 4.7: The minimal binary cyclic graphical model $\mathcal{K}^3$. LEFT: The minimal binary cyclic graphical model $\mathcal{K}^3 = (\mathcal{V}, \mathcal{E}) = (\{1,2,3\}, \{12,13,23\})$. RIGHT: The 8 vertices (white background) of the minimally represented marginal polytope $\mathcal{M}_{\mathcal{K}^3} \subset \mathbb{R}^6_+$ and the 4 additional non-integer vertices (red background) of the minimally represented local polytope $\mathcal{L}_{\mathcal{K}^3} \subset \mathbb{R}^6_+$.

The constraints are more conveniently stated using the so-called *minimal representation* of binary graphical models [74, Sect. 3.2], that involves the variables[1]

$$\mu_i := \mu_i(1),\ i \in \mathcal{V}, \qquad \mu_{ij} := \mu_i(1)\mu_j(1),\ ij \in \mathcal{E} \tag{4.47}$$

and encodes the local vectors (2.41) by

$$\begin{pmatrix} 1-\mu_i \\ \mu_i \end{pmatrix} \leftarrow \begin{pmatrix} \mu_i(0) \\ \mu_i(1) \end{pmatrix}, \qquad \begin{pmatrix} (1-\mu_i)(1-\mu_j) \\ (1-\mu_i)\mu_j \\ \mu_i(1-\mu_j) \\ \mu_{ij} \end{pmatrix} \leftarrow \begin{pmatrix} \mu_{ij}(0,0) \\ \mu_{ij}(0,1) \\ \mu_{ij}(1,0) \\ \mu_{ij}(1,1) \end{pmatrix}. \tag{4.48}$$

[1] We reuse the symbol μ for simplicity and only "overload" in this subsection the symbols μ_i, μ_{ij} for local vectors (2.41) by the variables on the left-hand sides of (4.47)

Thus, it suffices to use a single variable μ_i for every node $i \in \mathcal{V}$ instead of two variables $\mu_i(0), \mu_i(1)$, and also a single variable μ_{ij} for every edge $ij \in \mathcal{E}$ instead of four variables $\mu_{ij}(0,0), \mu_{ij}(0,1), \mu_{ij}(1,0), \mu_{ij}(1,1)$. Then the *local polytope constraints* (2.41) take the form

$$0 \leq \mu_{ij}, \qquad \mu_{ij} \leq \mu_i, \qquad \mu_{ij} \leq \mu_j, \qquad \mu_i + \mu_j - \mu_{ij} \leq 1, \qquad \forall ij \in \mathcal{E}. \tag{4.49}$$

The *marginal polytope constraints* additionally involve the so-called triangle inequalities [23]

$$\sum_{i \in \mathcal{V}} \mu_i - \sum_{jk \in \mathcal{E}} \mu_{jk} \leq 1, \tag{4.50a}$$

$$\mu_{12} + \mu_{13} - \mu_{23} \leq \mu_1, \qquad \mu_{12} - \mu_{13} + \mu_{23} \leq \mu_2, \qquad -\mu_{12} + \mu_{13} + \mu_{23} \leq \mu_3. \tag{4.50b}$$

The right panel of Figure 4.7 lists the 8 vertices of $\mathcal{M}_{\mathcal{K}^3}$ and the 4 additional vertices of $\mathcal{L}_{\mathcal{K}^3}$ that arise by dropping the subset of constraints (4.50).

We evaluate 10^5 models generated by randomly sampling the model parameters (2.35): With $\mathcal{U}[a,b]$ denoting the uniform distribution on the interval $[a,b] \subset \mathbb{R}$, we set

$$\theta_i = \begin{pmatrix} 1-p \\ p \end{pmatrix} - \frac{1}{2}\begin{pmatrix} 1 \\ 1 \end{pmatrix}, \quad p \sim \mathcal{U}[0,1], \qquad \theta_{ij} = \begin{pmatrix} p_1 & p_2 \\ p_3 & p_4 \end{pmatrix}, \quad p_i \sim \mathcal{U}[-2,2],\ i \in [4]. \tag{4.51}$$

Note the different scale, $\theta_i \in [-\frac{1}{2}, +\frac{1}{2}]^2$, $\theta_{ij} \in [-2,+2]^{2\times 2}$, which results in a larger influence of the pairwise terms and hence make inference more difficult. Suppose, for example, that the diagonal terms of θ_{ij} are large, which favours the assignment of *different* labels to the nodes $1,2,3 \in \mathcal{V}$. Then assigning labels 0 and 1 to the vertices 1 and 2, respectively, will inherently lead to a large energy contribution due to the assignment to node 3, no matter if this third label is 0 or 1, because it must agree with the assignment either to node 1 or to 2.

Every *binary* vertex listed by Fig. 4.7 (right panel) is the global optimum of both the LP relaxation (2.42) and the original objective function (4.2) in approximately $\approx 11.94\%$ of the 10^5 scenarios, whereas every *non-binary* vertex is optimal in approximately $\approx 1.12\%$.

An example where a *non-binary* vertex is optimal for the LP relaxation (2.42) is given by the model parameter values

$$\begin{aligned} \theta_1 &= \begin{pmatrix} -0.2261 \\ 0.2261 \end{pmatrix}, & \theta_{12} &= \begin{pmatrix} -0.9184 & -1.6252 \\ -1.8891 & -0.9807 \end{pmatrix}, \\ \theta_2 &= \begin{pmatrix} -0.4449 \\ 0.4449 \end{pmatrix}, & \theta_{13} &= \begin{pmatrix} 0.3590 & 0.0958 \\ -1.8668 & 1.5193 \end{pmatrix}, \\ \theta_3 &= \begin{pmatrix} -0.3202 \\ 0.3202 \end{pmatrix}, & \theta_{23} &= \begin{pmatrix} 1.2147 & -1.5215 \\ -0.3302 & -0.0459 \end{pmatrix}. \end{aligned} \tag{4.52}$$

The corresponding solutions of the marginal polytope $\mathcal{M}_{\mathcal{K}^3}$, the local polytope $\mathcal{L}_{\mathcal{K}^3}$ and our method are listed in Table 4.1. Since the LP-relaxation returns a non-binary solution, rounding in a post-processing step amounts to random guessing. In contrast, our method is able to determine the optimal solution because rounding is smoothly integrated into the overall optimization process.

		μ_1	μ_2	μ_3	Iterations
Marginal Polytope $\mathcal{M}_{\mathcal{K}^3}$		1	0	0	-
Local Polytope $\mathcal{L}_{\mathcal{K}^3}$		0.5	0.5	0.5	-
Our Method ($\tau = \frac{1}{10}$)	$\alpha = 0.2$	0.999	0.258e−3	0.205e−3	108
	$\alpha = 0.5$	0.999	0.161e−3	0.114e−4	14
	$\alpha = 0.9$	0.999	0.239e−4	0.546e−6	8

Table 4.1: Minimal cyclic graphical model on $\mathcal{K}^3$: Solutions $\mu = (\mu_1, \mu_2, \mu_3)$ of the marginal polytope $\mathcal{M}_{\mathcal{K}^3}$, the local polytope $\mathcal{L}_{\mathcal{K}^3}$ and our method. We use (4.52) as the parameter values for the triangle model. Our method was applied with threshold 10^{-3} as termination criterion (3.55), step-size $h = 0.5$, smoothing parameter $\tau = 0.1$ and three values of the rounding parameter $\alpha \in \{0.2, 0.5, 0.9\}$. By definition, minimizing over the marginal polytope returns the globally optimal discrete solution. The LP relaxation has a fractional solution for this model, so that rounding in a post-processing step amounts to random guessing. Our approach returns the global optimum in each case up to numerical precision.

Fig. 4.8 presents the results of the experiments for the minimal cyclic graphical model $\mathcal{K}^3$. In order to assess the influence of the *rounding* parameter α and the *smoothing* parameter τ, we evaluate all 10^5 models for *each pair* of (τ, α), with

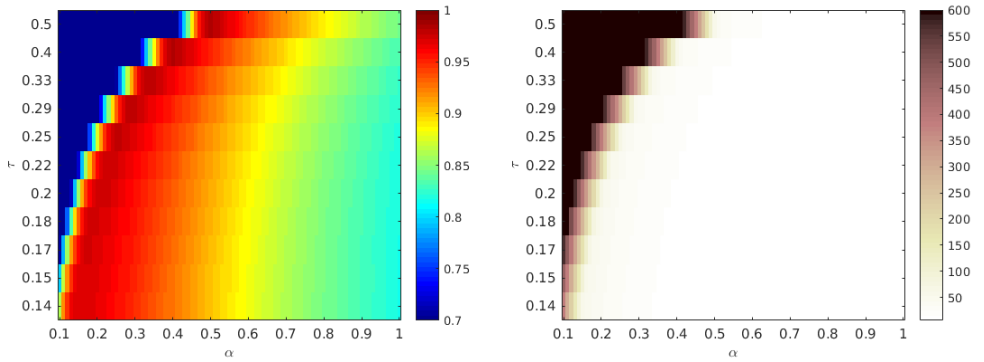


Figure 4.8: Evaluation of the minimal cyclic graphical model $\mathcal{K}^3$. For every pair of parameter values (τ, α), we evaluated 10^5 models, which were generated as explained in the text. In each experiment, we terminated the algorithm when the average entropy dropped below 10^{-3} or if the maximum number of 600 iterations was reached. In addition, we chose a constant step-size $h = 0.5$. LEFT: The plot shows the percentage of experiments where the energy returned by our algorithm had a relative error smaller then 1% compared to the minimal energy of the globally optimal integral labeling. In agreement with Fig. 4.5 (bottom), less aggressive rounding yielded labelings closer to the global optimum. RIGHT: This plot shows the corresponding average number of iterations. The black region indicates experiments where the maximum number of 600 iterations was reached, because too strong smoothing of the Wasserstein distance (large τ) overcompensated the effect of rounding (small α), so that the convergence criterion (3.55) which measures the distance to integral solutions, cannot be satisfied. In the remaining large parameter regime, the choice of α enables to control the trade-off between high-quality (low-energy) solutions and computational costs.

$\tau \in \{\frac{1}{2}, \frac{1}{2.5}, \ldots, \frac{1}{6.5}, \frac{1}{7}\}$ and $\alpha \in \{0.1, 0.11, \ldots, 0.99, 1\}$. These statistics show that our algorithm converges to integral solutions, except for very unbalanced parameter values: strong smoothing with large τ, weak rounding with small α. Within the remaining broad parameter regime, parameter α enables us to control the influence of rounding. In particular, in agreement with Fig. 4.5 (bottom), less aggressive rounding computes labelings closer to the global optimum.

Tab. 4.2 displays the success rate and the number of iterations for three different parameter configurations from Fig. 4.8. For instance, using $\alpha = 0.22$ and $\tau = 0.2$, our algorithm finds in 97.35% of the experiments an energy with relative error smaller

then 1% with respect to the optimal energy. In addition, the algorithm requires on average 45 iterations to converge. By using instead more aggressive rounding ($\alpha = 0.58$ and $\tau = 0.15$) in each iteration step (4.44), the average number of iterations reduces to 9, but the accuracy also drops down to 88.6%.

Overall, these experiments clearly demonstrate the ability to control the trade-off between high-quality (low energy) labelings and computational costs in terms of α, for all values of τ below a reasonably large upper bound. In addition, a small number of iterations is required to converge depending on the *rounding* parameter α.

α	τ	Success rate	Iterations
0.22	0.2	97.35%	45
0.5	0.33	93.41%	15
0.58	0.15	88.6%	9

Table 4.2: Minimal cyclic graphical model on $\mathcal{K}^3$: Three different parameter configurations and corresponding results. The table shows three different parameter configurations extracted from Fig. 4.8. The comparison of the success rate and the number of iterations until convergence clearly demonstrates the trade-off between accuracy of optimization and convergence rate, depending on the *rounding* parameter α and the *smoothing* parameter τ. Overall, the number of iterations is significantly smaller than for first-order methods of convex programming for solving the LP relaxation, that additionally require rounding as a post-processing step to obtain an integral solution.

4.5.3 Comparison to Other Methods

In this section we compare our geometric approach to established inference algorithms, namely *sequential tree-reweighted message passing* (TRWS) [42] and *loopy belief propagation* (Loopy-BP) [76] based on the OpenGM package [5].

For this comparison, we consider the noisy binary labeling scenario depicted by Fig. 4.9 (TOP ROW). Let $f\colon \mathcal{V} \to [0,1]$ denote the noisy image data given on the grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a 4-neighborhood and $\mathcal{X} = \{0,1\}$ denote the labels. We use the following data term and Potts prior,

$$\theta_i = \begin{pmatrix} f(i) \\ 1 - f(i) \end{pmatrix} \quad \text{for} \quad i \in \mathcal{V} \quad \text{and} \quad \theta_{ij} = \begin{pmatrix} 0 & 1 \\ 1 & 0 \end{pmatrix} \quad \text{for} \quad ij \in \mathcal{E}\,. \tag{4.53}$$

The threshold 10^{-3} is used for the normalized average entropy termination criterion (3.55). Figure 4.9 shows the visual reconstruction as well as the corresponding

discrete energy values and percentage of correct labels for all three methods. Our method has similar accuracy and returns a slightly better optimal discrete energy level than TRWS and Loopy-BP.

We investigate again the influence of the *rounding* mechanism by repeating the same experiment, but using different values of the rounding parameter $\alpha \in \{0.1, 1, 2, 5\}$. As shown by Fig. 4.9, the results confirm the findings of the preceding experiment: More aggressive rounding (α large) leads to faster convergence but yields less regularized results with higher energy values.

We wish to point out that the listed runtimes in Fig. 4.9 highly depend on the respective implementation. Our implementation was neither optimized nor parallelized. Parallelizing our approach is relatively simple, unlike TRWS which is sequential by design.

4.5.4 Non-Potts Prior

This experiment demonstrates that pre-specified pairwise model parameters (regularization) of a graphical model are properly taken into account by our approach. We apply our approach based on a non-Potts prior to a non-binary labeling problem with noisy input data, as depicted by Fig. 4.10.

As labels we use the following colors

$$\mathcal{X} = \{\ell_1 = \blacksquare, \ell_2 = \blacksquare, \ell_3 = \blacksquare, \ell_4 = \blacksquare, \ell_5 = \blacksquare\} \subset [0,1]^3\,, \tag{4.54}$$

which correspond to the five RGB-colors of the original image (Fig. 4.10). Let $f : \mathcal{V} \to [0,1]^3$ denote the noisy input image (Fig. 4.10, TOP ROW, center panel) given on the grid graph $\mathcal{G} = (\mathcal{V}, \mathcal{E})$ with a 4-neighborhood. This image is created by randomly selecting 40% of the original image pixels and uniformly sampling a label at those positions. As unary term we use the $\|\cdot\|_1$ distance with scaling factor $\rho > 0$

$$\theta_i = \frac{1}{\rho}\big(\|f(i) - \ell_1\|_1, \ldots, \|f(i) - \ell_5\|_1\big), \quad i \in \mathcal{V}. \tag{4.55}$$

Assuming we had prior knowledge of the image labeling problem. For example, let the RGB-colors encode the *image direction* of the respective pixel, i.e. $\blacksquare$ = "top", $\blacksquare$ = "bottom", $\blacksquare$ = "center", $\blacksquare$ = "left", and $\blacksquare$ = "right" (Fig. 4.10 TOP ROW, left). Hence, we know in advance that it makes no sense if "top" and "bottom" as well as "left" and "right" are adjacent to each other, because they are separated by the "center". This prior knowledge can be taken into account by specifying the *non-*

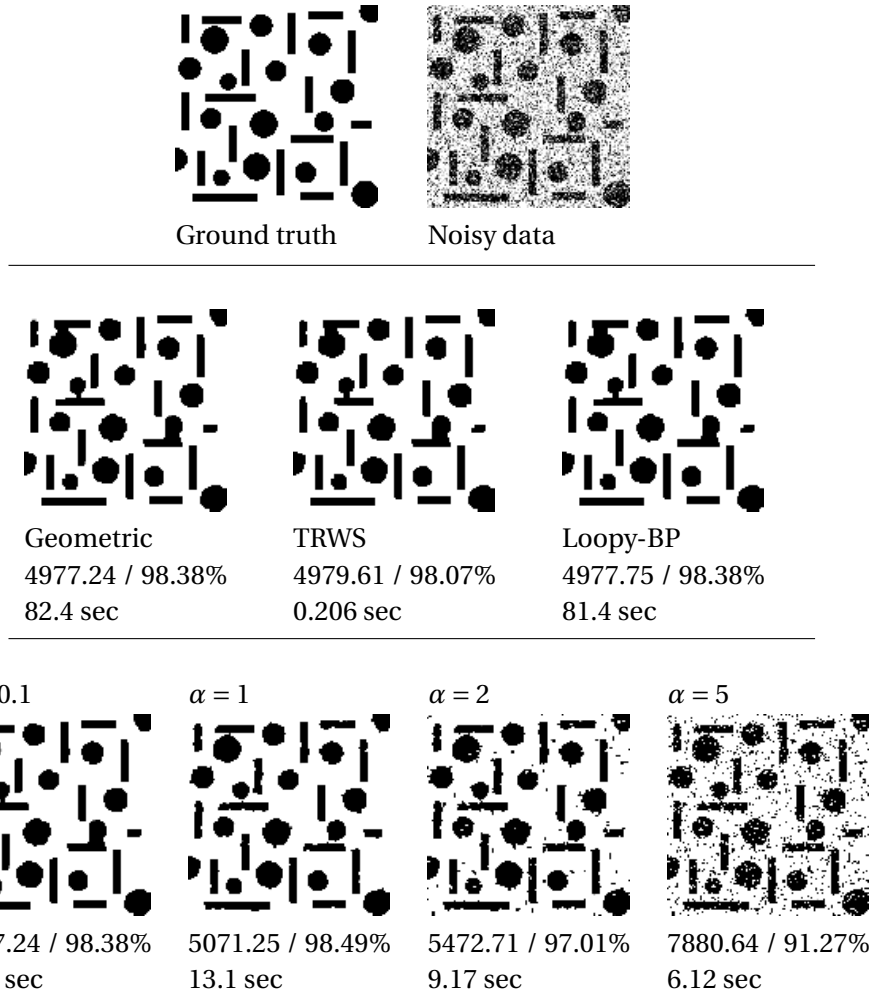


Figure 4.9: Comparison to other methods. TOP ROW: Noisy image labeling problem: a binary ground truth image (LEFT) to be recovered from noisy input data (RIGHT). MIDDLE ROW: Results for the noisy labeling problem using a standard data term and Potts prior (4.4) with *discrete energy* / *accuracy* / *runtime*. Parameter values for the geometric approach: smoothing $\tau = 0.1$, step-size $h = 0.2$ and rounding strength $\alpha = 0.1$. The threshold for the termination criterion is 10^{-3}. All methods show similar performance. BOTTOM ROW: Labeling results of the geometric approach using different values of the rounding parameter $\alpha \in \{0.1, 1, 2, 5\}$ with *discrete energy* / *accuracy* / *runtime*: more aggressive rounding (α large) leads to less regularized results with higher energy values. Parameter values of the geometric approach: smoothing $\tau = 0.1$, step size $h = 0.2$ and threshold 10^{-3} for termination.

uniform pairwise term as follows

$$\theta_{ij} = \frac{1}{10}\begin{pmatrix} 0 & 10 & 1 & 1 & 1 \\ 10 & 0 & 1 & 1 & 1 \\ 1 & 1 & 0 & 1 & 1 \\ 1 & 1 & 1 & 0 & 10 \\ 1 & 1 & 1 & 10 & 0 \end{pmatrix}, \qquad ij \in \mathcal{E}, \tag{4.56}$$

which penalizes the unlikely label transitions by a factor of 10. In words, the entries of θ_{ij} with a penalty value 1 correspond to the *unlikely* label transitions $\ell_1 \leftrightarrow \ell_2$ and $\ell_4 \leftrightarrow \ell_5$, whereas all other *natural* transitions are endowed with smaller penalty values of 0 and 0.1, respectively.

The unlikely transitions $\ell_1 \leftrightarrow \ell_2$ ($\leftrightarrow$) and $\ell_4 \leftrightarrow \ell_5$ ($\leftrightarrow$) can be easily confused by the unary term (4.55), due to the small distance of the colors representing these labels. We would like to emphasize that no color embedding is used to facilitate this regularization task. The prior knowledge about the labeling problem is exclusively used in the definition of the non-uniform prior (4.56) which was considered as *given* in terms of some discrete graphical model.

We show the influence of these non-uniform parameters on the labeling by comparing this model with a model where the pairwise terms are replaced by a *Potts prior* (4.4) with $\lambda = \frac{1}{10}$. We use the scaling factor $\rho = 15$ for the unaries, a constant step-size $h = 0.1$, rounding parameter $\alpha = 0.01$, smoothing parameter $\tau = 0.01$ and 10^{-4} as threshold for the normalized average entropy termination criterion (3.55).

The results depicted in Fig. 4.10 (BOTTOM ROW) clearly show the positive influence of the non-Potts prior (labeling accuracy 99.34%) whereas using the Potts prior lowers the accuracy to 87.12%. This is due to the fact that the color labels ℓ_1 and ℓ_2 as well as ℓ_4 and ℓ_5 have a relatively small $\|\cdot\|_1$ distance and are therefore not easy to distinguish using both the data term and a Potts prior. On the other hand, the additional knowledge about valid label configurations encoded by the non-uniform prior (4.56) is sufficient to overcome this difficulty, despite using the same data term, and to separate the regions correctly.

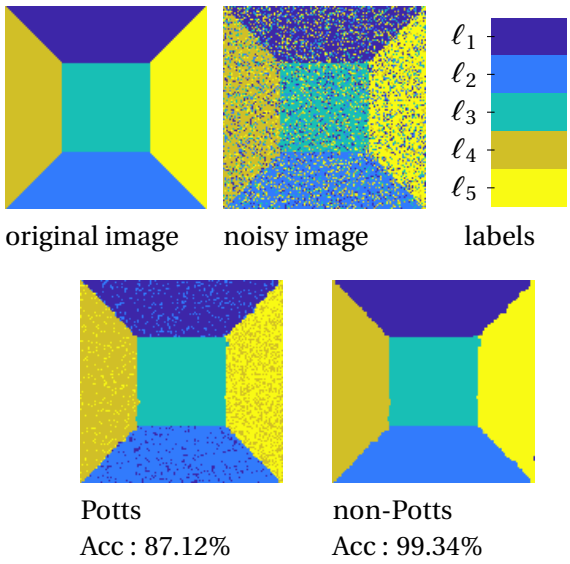


Figure 4.10: Non-Potts prior example. TOP ROW: Original image (left), encoding the image directions "top", "bottom", "center", "left" and "right" by the RGB-color labels $\ell_1, \ell_2, \ell_3, \ell_4$ and ℓ_5 (right). The noisy test image (middle) is created by randomly selecting 40% of the original image pixels and uniformly sampling a label at those positions. Unlikely label transitions $\ell_1 \leftrightarrow \ell_2$ and $\ell_4 \leftrightarrow \ell_5$ are represented by color (feature) vectors that are close to each other and hence can be easily confused. BOTTOM ROW: Results of the labeling problem using the Potts and non-Potts prior model together with *accuracy values* (Acc). Parameters for this experiment are $\rho = 15$, smoothing $\tau = 0.01$, step-size $h = 0.1$ and rounding strength $\alpha = 0.01$. The threshold for the termination criterion (3.55) is 10^{-4}.

Chapter 5

Model Parameter Learning for Adaptive Regularization

The focus of the present chapter is on the inverse problem of inference, the model parameter learning problem. As discussed in Section 2.2.3, learning the canonical parameters of a discrete graphical model with an underlying cyclic graph is quite challenging for the following reasons.

- The evaluation of the partition function (2.23) is intractable and must be approximated.
- The inference subroutine of the learning framework can only be carried out approximatively.

Since these *approximation errors* definitely influence the effectiveness of the learning procedure, we take another way by learning the parameters for the assignment flow directly. Thereby, we ignore the connection to discrete graphical models derived in the previous chapter. In contrast to graphical models, this strategy has the benefit that the underlying inference task is solved *exactly* by following the assignment flow which drives the learning process of the parameters.

Our proposed problem formulation (Section 5.1) has the convenient property that the solution space of the parameters has again the form of an assignment manifold (Section 5.1.1). Thus, all definitions, maps and integration schemes derived in Chapter 3 can directly be transferred and applied. Afterwards, we provide the expressions of the corresponding gradients and differentials (Section 5.2) and explain our optimization strategy (Section 5.3). We end this chapter by numerically evaluating our approach with two different types of experiments (Section 5.4).

The chapter is based on joint work with Fabrizio Savarino, Stefania Petra and Christoph Schnörr, that was published as a conference paper [35] and as a more detailed journal paper [37].

5.1 Problem Formulation

The parameter learning problem (5.1) is a specific instance of the general parameter estimation formulation (1.4). The goal is to adapt the weights (3.39) of the linear

assignment flow (3.49) so as to preserve important image structure in a supervised manner. More specifically, we propose the following problem

$$\min_{\Omega \in \mathcal{P}} \quad \mathcal{C}\big(V(T,\Omega)\big) \tag{5.1a}$$

$$\text{s.t.} \quad \dot{V}(t) = F(V(t),\Omega), \quad t \in [0,T], \quad V(0) = 0, \tag{5.1b}$$

with components

- $\mathcal{P}$ denotes the *parameter manifold* which represents the weights ω_{ik} from (3.39), in the following collectively denoted by Ω (see Section 5.1.1).
- $\mathcal{C}$ is a *objective function* which measures the discrepancy between ground truth labeling W^* and the labeling induced by $V(T) = V(T,\Omega)$ at fixed time T (see Section 5.1.2).
- $F(V,\Omega)$ denotes the vector field generating the *modified linear assignment flow* (see Section 5.1.3).

Remark 5.1 It is important to note that the dependency of $\mathcal{C}(V(T,\Omega))$ on the weights Ω is only *implicitly* given through the solution $V(T) = V(T,\Omega)$ of the flow (5.1b). Therefore, we present in Section 5.3 a numerical first-order scheme for optimizing (5.1) where the gradient of $\mathcal{C}(V(T,\Omega))$ with respect to the parameter Ω is calculated using the sensitivity analysis from Section 2.3. △

5.1.1 The Parameter Manifold

The parameter manifold represents the weights ω_{ik} from (3.39) associated to the neighborhood $\mathcal{N}_i$, $i \in \mathcal{V}$. To simplify the exposition, we assume that all neighborhoods $\mathcal{N}_i$ have the same size

$$N := |\mathcal{N}_i| \quad \text{for all } i \in \mathcal{V}. \tag{5.2}$$

Due to the constraints (3.39), the weight vector $\Omega_i := (\omega_{i1},\dots,\omega_{iN})^\top$ can be viewed as a point in $\mathcal{S}_N$. Accordingly, we define the *parameter manifold*

$$\mathcal{P} := \underbrace{\mathcal{S}_N \times \ldots \times \mathcal{S}_N}_{m\text{-times}} \tag{5.3}$$

as feasible set for learning the weights, which has the form of an assignment manifold (3.18). Following the line of Section 3.1.2, $\mathcal{P}$ becomes a Riemannian manifold

$(\mathcal{P}, g)$ with the Fisher-Rao metric g and is identified with the embedding into $\mathbb{R}^{m\times N}$

$$\mathcal{P} = \{\Omega \in \mathbb{R}^{m\times N} \colon \Omega \mathbb{1}_N = \mathbb{1}_m \text{ and } \Omega_{ik} > 0 \text{ for all } i \in [m], k \in [N]\}. \tag{5.4}$$

Each point $\Omega \in \mathcal{P}$ represents a global choice of weights with Ω_i representing the weights ω_{ik} associated to the neighborhood $\mathcal{N}_i$ in (3.39). The constant tangent space of $\mathcal{P}$ is denoted by $\mathcal{T}_\mathcal{P}$ and the corresponding orthogonal projection by

$$\Pi_\mathcal{P} \colon \mathbb{R}^{m\times N} \to \mathcal{T}_\mathcal{P}, \quad M \mapsto \Pi_\mathcal{P}[M] = (\Pi_{\mathcal{T}_N}[M_1], \dots, \Pi_{\mathcal{T}_N}[M_m])^\top. \tag{5.5}$$

The global uniform weights are given by the *barycenter*

$$\mathbb{1}_\mathcal{P} := (\mathbb{1}_{\mathcal{S}_N}, \dots, \mathbb{1}_{\mathcal{S}_N}) = \mathbb{1}_m \mathbb{1}_{\mathcal{S}_N}^\top \in \mathcal{P}, \tag{5.6}$$

where the second equality is due to the embedding (5.4).

Remark 5.2 Based on this parametrization, we compute a global expression of the differential $dS(W_0)$ and describe the linear assignment flow (3.50) on the tangent space by a corresponding expression in (5.15). △

5.1.2 Objective Function

In the supervised setting, an image and a corresponding ground truth labeling are given. We denote this ground truth labeling by W^* where every row W_i^* is some unit basis vector e_{k_i} of $\mathbb{R}^n$ representing the ground truth label ℓ_{k_i} at node $i \in \mathcal{V}$. In addition, the state $V \in \mathcal{T}_\mathcal{W}$ of the assignment flow (5.1b) parametrizes an assignment $W = \exp_{\mathbb{1}_\mathcal{W}}(V) \in \mathcal{W}$.

Our objective function consists of accumulating the KL-divergence between the ground truth W_i^* and the assignment W_i for every node $i \in \mathcal{V}$,

$$\mathrm{KL}(W_i^*, W_i) = \sum_{j\in[n]} W_{ij}^* \log\left(\frac{W_{ij}^*}{W_{ij}}\right) = \langle W_i^*, \log(W_i^*)\rangle - \langle W_i^*, \log(W_i)\rangle, \tag{5.7}$$

which results in a measure of the global deviation between W parametrized by V and the ground truth W^*

$$\mathcal{C}(V) := \sum_{i\in\mathcal{V}} \mathrm{KL}(W_i^*, \exp_{\mathbb{1}_{\mathcal{S}_n}}(V_i)) = \langle W^*, \log(W^*)\rangle - \langle W^*, \log\big(\exp_{\mathbb{1}_\mathcal{W}}(V)\big)\rangle. \tag{5.8}$$

Remark 5.3 Again, $\mathcal{C}$ does *not explicitly* depend on the weights $\Omega \in \mathcal{P}$. In the problem formulation (5.1a), this dependency is only given *implicitly* through the evaluation of $\mathcal{C}$ at $V(T,\Omega)$, where $V(T,\Omega)$ depends on the parameters Ω as solution of the modified linear assignment flow (5.9).

5.1.3 Modified Linear Assignment Flow

In our supervised formulation (5.1), the data is represented by the likelihood matrix (3.41). Depending on the initial choice of weights Ω_0, the similarity matrix (3.42) comprises the averaged data information. Hence, the data only influences the linear assignment flow (3.50) through the constant similarity matrix $S(W_0)$. However, since the initial weights Ω_0 are in general not adapted to any specific image structure, this can lead to a loss of desired structural information through $S(W_0)$ at the outset, that cannot be recovered afterwards.

To avoid this problem, we slightly *modify* the linear assignment flow (3.50) to obtain an explicit *data term*, independent of the choice of initial weights. This is done by replacing the constant term $S(W_0)$ with the lifted distances $L(W_0)$, which results in the *modified linear assignment flow*

$$\dot{V} = \Pi_{\mathcal{T}_\mathcal{W}}\big[L(W_0)\big] + dS(W_0)[V], \quad V(0) = 0, \quad W(t) = \exp_{\mathbb{1}_\mathcal{W}}\big(V(t)\big). \tag{5.9}$$

Remark 5.4 We wish to point out that the similarity matrix $S(W_0)$ is also involved in the expression $R_{S(W_0)}$ of the differential $dS(W_0)$ (see (5.14) below). However, the effect of this with respect to the initial weights is negligible. By keeping $S(W_0)$ constant the right hand side of (5.9) is *linear* with respect to both the tangent vector V and the parameters Ω. △

5.2 Gradients and Differentials

In Section 5.3 we calculate the gradient of $\mathcal{C}(V(T,\Omega))$ with respect to the parameters Ω by using the sensitivity analysis from Section 2.3. In view of the formula for the adjoint sensitivity (see Theorem 2.9 & 2.10), we need the following expressions that we collect in this section:

$\nabla\mathcal{C}$	*Euclidean gradient* of objective function (5.8) w.r.t. the *states* V.
$d_V F(V,\Omega)$	*differential* of the right-hand side (5.9) w.r.t. the *states* V.
$d_\Omega F(V,\Omega)$	*differential* of the right-hand side (5.9) w.r.t. the *parameters* Ω.

Proposition 5.1 (Euclidean gradient of $\mathcal{C}$)
The *Euclidean gradient* of objective (5.8) for fixed $W^* \in \mathcal{W}$ is given by

$$\nabla\mathcal{C}(V) = \exp_{\mathbb{1}_\mathcal{W}}(V) - W^* \quad \text{for} \quad V \in \mathcal{T}_\mathcal{W}. \tag{5.10}$$

Proof See Appendix A.4. □

In order to simplify the following formulas and calculations, we calculate a global expression for the differential $dS(W)$. To do so, we define the *averaging matrix* $A_\Omega \in \mathbb{R}^{m\times m}$ depending on the weights $\Omega \in \mathcal{P}$ by

$$(A_\Omega)_{ik} := \delta_{k\in\mathcal{N}_i}\Omega_{ik} = \begin{cases} \Omega_{ik}, & \text{for } k \in \mathcal{N}_i \\ 0, & \text{else} \end{cases}, \tag{5.11}$$

where $\delta_{k\in\mathcal{N}_i}$ takes the value 1 if $k \in \mathcal{N}_i$ and 0 otherwise. Note that the averaging matrix A_Ω *linearly* depends on the weight parameters Ω. Thus, A_Ω parametrizes averages with respect to the underlying graph structure $\mathcal{G}$ in terms of the given the neighborhoods (3.37). Then, the averages of the row vectors of a matrix $M \in \mathbb{R}^{m\times n}$ with weights Ω are given by the matrix multiplication $A_\Omega M$, with the i-th row vector given by

$$(A_\Omega M)_i = \sum_{k\in\mathcal{N}_i} \omega_{ik} M_k. \quad \text{for all } i \in \mathcal{V}. \tag{5.12}$$

For later use, we record the following formula for the adjoint of A_Ω as a linear map with respect to Ω.

Lemma 5.2
If the averaging matrix is viewed as a linear map $A\colon \mathbb{R}^{m\times N} \to \mathbb{R}^{m\times m}$, $\Omega \mapsto A_\Omega$, then the adjoint map $A^\top\colon \mathbb{R}^{m\times m} \to \mathbb{R}^{m\times N}$, $B \mapsto A_B^\top$ is given by

$$\left(A_B^\top\right)_{ij} = B_{ij} \quad \text{for } i \in \mathcal{V}, j \in \mathcal{N}_i. \tag{5.13}$$

Proof See Appendix A.4. □

By plugging A_Ω (5.11) into the definition of the differential (3.51) we obtain the global expression

$$dS(W)[V] = R_{S(W)}\left[A_\Omega\left(\frac{V}{W}\right)\right], \quad \text{for all } V \in \mathcal{T}_\mathcal{W},\ W \in \mathcal{W}, \tag{5.14}$$

for the differential of the similarity matrix (3.42). Accordingly, the linear assignment flow (5.9) on the vector space $\mathcal{T}_\mathcal{W}$ takes the form

$$\dot V(t) = \Pi[L(W_0)] + R_{S(W_0)}\left[A_\Omega\left(\frac{V(t)}{W_0}\right)\right], \quad V(0) = 0, \quad W(t) = \exp_{\mathbb{1}_\mathcal{W}}(V(t)). \tag{5.15}$$

Remark 5.5 Equation (5.15) highlights the importance to fix $S(W_0)$ in order to obtain a model that is *linear* in both the state vector V and the parameters Ω. △

The following proposition collects the differentials of the right hand side of (5.15).

Proposition 5.3 (Differentials of the right hand side F)
The differentials of the map $F\colon \mathcal{T}_\mathcal{W} \times \mathcal{P} \to \mathcal{T}_\mathcal{W}$ of the right-hand side of (5.9) with respect to the first and second argument are given by

$$d_V F(V,\Omega)\colon \mathcal{T}_\mathcal{W} \to \mathcal{T}_\mathcal{W}, \qquad X \mapsto d_V F(V,\Omega)[X] = R_{S(W_0)}[A_\Omega X], \tag{5.16a}$$
$$d_\Omega F(V,\Omega)\colon \mathcal{T}_\mathcal{P} \to \mathcal{T}_\mathcal{W}, \qquad \Psi \mapsto d_\Omega F(V,\Omega)[\Psi] = R_{S(W_0)}[A_\Psi V]. \tag{5.16b}$$

The corresponding adjoint mappings with respect to the standard Euclidean structure of $\mathbb{R}^{m\times n}$ are

$$d_V F(V,\Omega)^\top\colon \mathcal{T}_\mathcal{W} \to \mathcal{T}_\mathcal{W}, \qquad X \mapsto d_V F(V,\Omega)^\top[X] = A_\Omega^\top R_{S(W_0)}[X], \tag{5.17a}$$
$$d_\Omega F(V,\Omega)^\top\colon \mathcal{T}_\mathcal{W} \to \mathcal{T}_\mathcal{P}, \qquad X \mapsto d_\Omega F(V,\Omega)^\top[X] = \Pi_\mathcal{P}\left[A^\top_{(R_{S(W_0)}[X])V^\top}\right], \tag{5.17b}$$

with the adjoint $A^\top_{(\cdot)}$ from Lemma 5.2.

Proof See Appendix A.4. □

5.3 Numerical Optimization

Summarizing the previous sections, the optimization problem (5.1) for adapting the weights of the modified linear assignment flow (5.9) takes explicitly the form

$$\min_{\Omega \in \mathcal{P}} \quad \sum_{i \in \mathcal{V}} \mathrm{KL}\big(W_i^*, W_i(T)\big) \quad \text{with} \quad W(T) = \exp_{\mathbb{1}_{\mathcal{W}}}\big(V(T)\big) \tag{5.18a}$$

$$\text{s.t.} \quad \dot{V}(t) = \Pi[L(W_0)] + R_{S(W_0)}\left[A_\Omega\left(\frac{V}{W_0}\right)\right], \quad t \in [0,T], \quad V(0) = 0. \tag{5.18b}$$

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

$$\Phi \colon \mathcal{P} \to \mathbb{R}, \quad \Omega \mapsto \Phi(\Omega) := \mathcal{C}(V(T,\Omega)) = \sum_{i \in \mathcal{V}} \mathrm{KL}(W_i^*, W_i(T,\Omega)). \tag{5.19}$$

Due to (3.27), the Riemannian gradient flow on $\mathcal{P}$ takes the form

$$\dot{\Omega}(t) = -\operatorname{grad}_{\mathcal{P}} \Phi\big(\Omega(t)\big) = -R_\Omega\big[\nabla\Phi\big(\Omega(t)\big)\big], \quad \text{with} \quad \Omega(0) = \mathbb{1}_{\mathcal{P}}, \tag{5.20}$$

where R_Ω is given by (3.25) and $\Omega(0) = \mathbb{1}_{\mathcal{P}}$ represents an *unbiased* initialization (5.6), i.e. *uniform* weights for every node $i \in \mathcal{V}$.

We use the geometric explicit Euler scheme (3.48) with constant step-size $h' > 0$ for numerically discretizing (5.20) (see Algorithm 5.1). A subroutine of this procedure is the computation of the *Euclidean gradient* $\nabla\Phi(\Omega)$ (see Algorithm 5.2) that we explain next. Since $\Phi(\Omega) = \mathcal{C}(V(T,\Omega))$ depends only *implicitly* on Ω given through the solution $V(t,\Omega)$ of the modified linear assignment flow (5.9), and according to (2.53), the gradient of Φ decomposes as

$$\nabla\Phi(\Omega) = d_\Omega V(T,\Omega)^\top \big[\nabla\mathcal{C}(V(T,\Omega))\big], \tag{5.21}$$

where $d_\Omega V(T,\Omega)^\top$ is the sensitivity of the solution $V(T,\Omega)$ with respect to Ω and $\nabla\mathcal{C}(V(T,\Omega))$ can be interpreted as an adjoint direction (see the paragraph below Theorem 2.5).

We determine the adjoint sensitivity (5.21), which drives the Riemannian gradient descent flow and in turn adapts the weights Ω, by choosing the *discretize-then-differentiate* approach (2.71). Recall the commutative diagram of Fig. 2.2 and relations summarized as Remark 2.7. We use an explicit Euler method with constant step-size $h > 0$, which results in Algorithm 5.2.

Algorithm 5.1: Explicit Euler discretization of the *Riemannian flow* (5.20).

Data: Initial weights $\Omega^{(0)} = \mathbb{1}_{\mathcal{P}}$, objective function $\Phi(\Omega) = \mathcal{C}\big(V(T,\Omega)\big)$, step-size h'
Result: Weight parameter estimates Ω^*
`// geometric Euler integration`
for $k = 0,\dots,K$ **do**
  compute $\nabla\Phi(\Omega^{(k)})$; `// Algorithm 5.2`
  $\Omega^{(k+1)} = \exp_{\Omega^{(k)}}\big(-h' R_{\Omega^{(k)}}\big[\nabla\Phi(\Omega^{(k)})\big]\big)$;

Algorithm 5.2: Computation of the Euclidean gradient $\nabla\Phi(\Omega^{(k)})$ (5.21).

Data: Current weights $\Omega^{(k)}$, step-size h
Result: Objective value $\Phi(\Omega^{(k)}) = \mathcal{C}(V^{(N)}(\Omega^{(k)}))$, adjoint sensitivity $\partial\Phi(\Omega^{(k)})$
`// forward Euler integration`
for $j = 0,\dots,N-1$ **do**
  $V^{(j+1)} = V^{(j)} + hF\big(V^{(j)},\Omega^{(k)}\big)$;
compute $\lambda^{(N)} = \nabla\mathcal{C}(V^{(N)}(\Omega^{(k)}))$;
set $\nabla\Phi(\Omega) = 0$;
`// backward Euler integration`
for $j = N-1,\dots,0$ **do**
  $\lambda^{(j)} = \lambda^{(j+1)} + h d_V F\big(V^{(j)},\Omega^{(k)}\big)^\top \lambda^{(j+1)}$;
  $\nabla\Phi(\Omega) \mathrel{+}= h d_\Omega F\big(V^{(j-1)},\Omega^{(k)}\big)^\top \lambda^{(j)}$; `// summand of (5.21)`

5.4 Experiments

In this section, we demonstrate and evaluate our approach using two types of experiments:

1. *Adaptive Regularization of Curvilinear Line Structures:* We consider a scenario with 3 labels and curvilinear line structure that has to be detected and labeled explicitly in noisy data. Just using uniform weights for regularization must fail. In addition to the noise, the actual image structure is randomly generated as well and defines a class of images. We demonstrate empirically that learning the weights to adapt within local neighborhoods from example data solves this problem (Section 5.4.1).

2. *Pattern Formation by Label Transport:* The second experiment adopts a different viewpoint and focuses on pattern formation, rather than on pattern de-

tection and recovery. We demonstrate the modeling expressiveness of the assignment flow with respect to pattern formation. In fact, by using the *linear assignment flow* as in the present chapter, label information can be flexibly transported across the image domain under certain conditions. These experiments just indicate what can be done, in principle, and stimulate further research directions (Section 5.4.2).

5.4.1 Adaptive Regularization of Curvilinear Line Structures

We consider a collection of images containing line structures induced by random Voronoi diagrams (Fig. 5.1, panel (a)). The goal is pixel-accurate labeling of any given image with three labels

$$\mathcal{X} = \{\ell_1 = \blacksquare, \ell_2 = \blacksquare, \ell_3 = \blacksquare\} \subset [0,1]^3 , \tag{5.22}$$

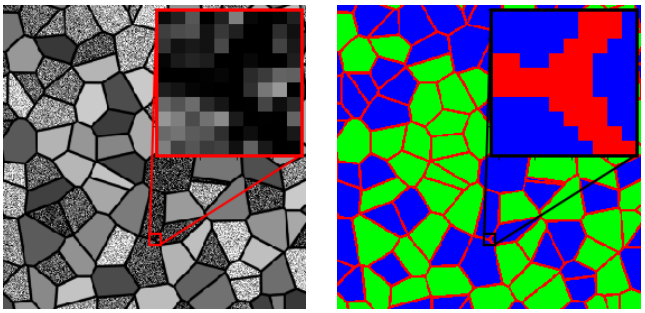
which represent $\blacksquare$ = "thin line structure", $\blacksquare$ = "homogeneous regions" and $\blacksquare$ = "texture". As usual in *supervised machine learning*, we first apply our approach during a *training phase* in order to learn weight adaptivity from ground truth labelings, and subsequently evaluate it in a *test phase* using *novel unseen data*.

Training Phase

We use 20 randomly generated images together with ground truth as *training data*: Figure 5.1(a) shows one of these images and Figure 5.1(b) the corresponding ground truth. Using these data we *learn how to adapt the regularization parameter* of the modified linear assignment flow (5.9) by solving problem (5.1), with the specific form given by (5.18).

Feature Vectors. The basis of our feature vectors are the outputs of simple 7×7 first- and second-order derivative filters, which are tuned to orientations at $0, 15, \ldots, 180$ degrees (we took absolute values of filter outputs to eliminate the $180 \sim 360$ degree symmetry). We reduce the dimension of the resulting feature vectors from 24 to 12 by taking the maximum of the first-order and second-order filter outputs, for each orientation. To incorporate more spatial information, we extract 3×3 patches from this 12-dimensional feature vector field. Thus, our *feature vectors* f_i, $i \in \mathcal{V}$ have dimension $3 \times 3 \times 12 = 108$ and are given as a point set in the Euclidean feature space $\mathcal{F} = \mathbb{R}^{108}$.

Label Extraction. Using ground truth information, we divide all feature vectors extracted from the training data into three classes: thin *line* structure, *homogeneous*



(a) input scene (b) ground truth

Figure 5.1: Training data. The training data consist of 20 pairs of randomly generated images: **(a)** an *input scene*, and **(b)** the corresponding *ground truth*. The ground truth images encode the labels with colors {■, ■, ■} = {line, homogeneous, texture}. Even though the global image structure can be easily assessed by the human eye, assigning correct labels pixelwise by an algorithm requires context-sensitive decisions, as the close-up view illustrates.

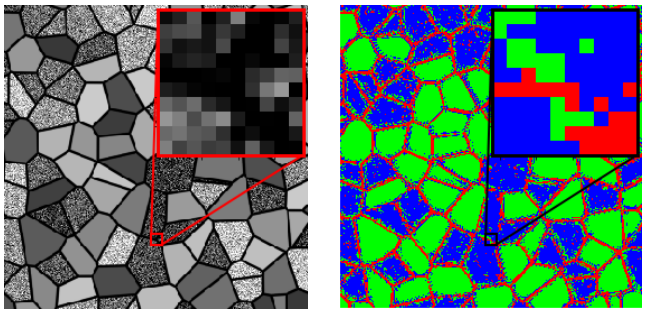
region and *texture*. We compute 200 prototypical feature vectors $\ell_{cj} \in \mathcal{F}$, $j \in [200]$, in each class $c \in \{\text{line}, \text{homogeneous}, \text{texture}\}$ by k-means clustering. Thus, each label (ℓ_1, ℓ_2, ℓ_3) of (5.22) is represented by 200 feature vectors in $\mathcal{F}$.

Distance Matrix. Even though in the original formulation of D (3.40) labels are represented by a single feature vector, multiple representatives can be taken into account as well by modifying the distance matrix (3.40) accordingly. Again, by using $c \in \{\text{line}, \text{homogeneous}, \text{texture}\} = \{1,2,3\}$, we define the entries of the distance matrix D_{ic}, for every $i \in \mathcal{V}$, as the distance between f_i and the best fitting representative ℓ_{cj} for class c, i.e.

$$D_{ic} := \min_{j \in [200]} \|f_i - \ell_{cj}\|_2. \tag{5.23}$$

The quality of this distance information is illustrated by Figure 5.2(b) that shows the labeling obtained by *local rounding*, i.e. by assigning to each pixel i the label $c = \min_{\tilde{c}} D_{i\tilde{c}}$. Although the result looks similar to the ground truth (see Fig. 5.1(b)), it is actually quite noisy when looking to single pixels in the close-up view of Figure 5.2(b).

Optimization. For each input image of the training set, we solve problem (5.1) using Algorithms 5.1 and 5.2 and the following parameter values: $|\mathcal{N}_i| = 9 \times 9$ (size



(a) random input data **(b) local rounding of the distances (5.23)**

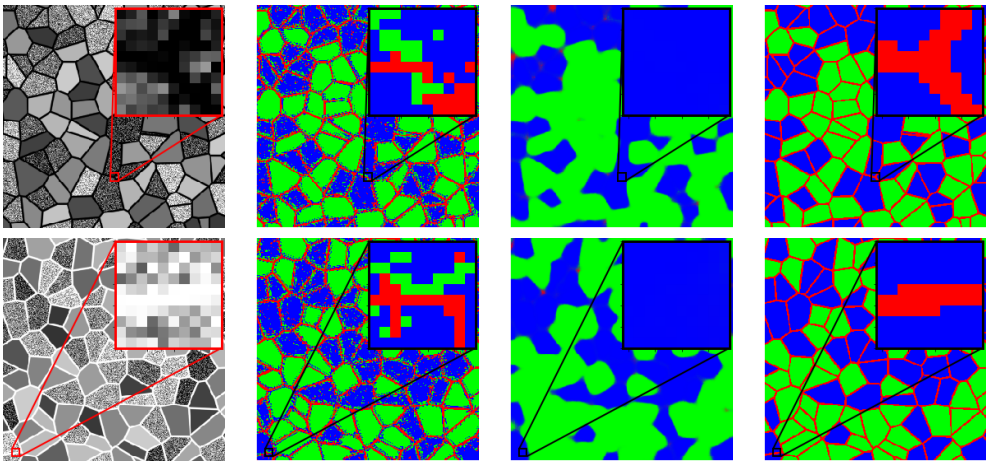
Figure 5.2: Input data and local label assignments. The plots illustrate the input data and the quality of the distances (5.23) between extracted feature vectors. Panel **(a)** shows a randomly generated input image from which features are extracted, as described in the text. Panel **(b)** shows the labeling obtained by local rounding, i.e. by assigning the label, that minimizes the corresponding distance, to each respective pixel. Comparing the close-up views of panel (b) and Fig. 5.1(b) (ground truth) shows that label assignments to individual pixels are noisy and incomplete. ({■, ■, ■} = {line, homogeneous, texture})

of local neighborhoods, for every i), $\rho = 1$ (scaling parameter for distance matrix, cf. (3.41)), $h = 0.5$ (constant step-size for computing the gradient with Alg. 5.2), and $T = 6$ (end of time horizon). As for numerical optimization on the parameter manifold $\mathcal{P}$ through the Riemannian gradient flow (Alg. 5.1), we use an initial value of $h' = 0.0125$ together with backtracking for adapting the step-size, for a maximal number of 100 iterations, and we terminate the iteration once the relative change

$$\frac{|\Phi(\Omega^{(k)}) - \Phi(\Omega^{(k-1)})|}{h'|\Phi(\Omega^{(k)})|} \tag{5.24}$$

of the objective function $\Phi\big(\Omega^{(k)}\big) = \mathcal{C}\big(V^{(N)}(\Omega^{(k)})\big)$ drops below 0.001.

Results. Figure 5.3 shows two results obtained during the training phase. They illustrate that *non-adaptive* regularization using *uniform* weights results in blurred partitions and fails completely to detect and label the line structures (panel (c)). On the other hand, the *adapted* regularizer preserves and restores the structure nearly

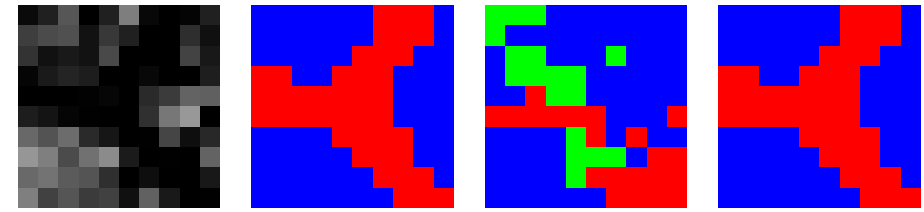


(a) input scene (b) *local rounding* of distance information (c) labeling with *uniform* weights (d) labeling with *optimal* weights

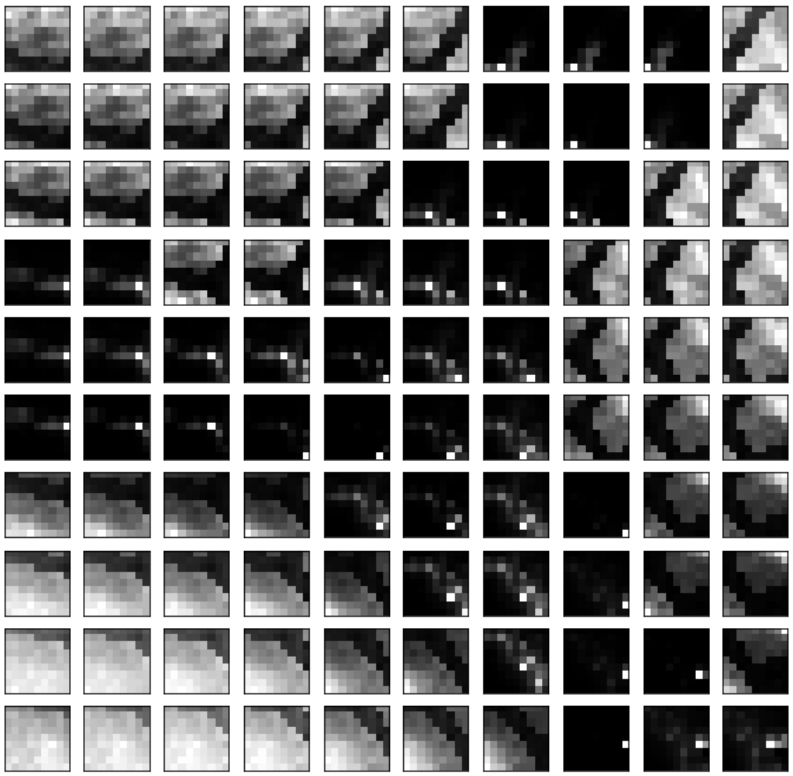
Figure 5.3: Training phase: Labeling results. This figure shows results of the training phase. Panel **(a)** shows the given input scene and panel **(b)** the corresponding *locally rounded* distance information. The labeling with *uniform* regularization (panel **(c)**) returns *smoothed over* regions and completely fails to preserve the line structures. The *adaptive* regularizer preserves the line structure nearly perfect (panel **(d)**), i.e. the *optimal weights* are able to steer the linear assignment flow successfully towards the given ground-truth labeling. ({■, ■, ■} = {line, homogeneous, texture})

perfect (panel (d)), i.e. the *optimal* weights steered the linear assignment flow towards the given ground-truth labeling.

Figure 5.4 shows a close-up view of a 10×10 pixel region together with the corresponding 10×10 *optimal weight patches*, extracted from Ω^*. The top row depicts (a) the training data, (b) the corresponding ground truth, (c) the local label assignments, and (d) the labeling obtained by using the learned weights Ω^*. Plot (e) shows the corresponding optimal weight patches $\Omega_i^* = (\omega_{i1}, \ldots, \omega_{iN})^\top$ associated to every pixel i in the 10×10 pixel region, where small and large weights are indicated by dark and bright gray values, respectively. These weight patches illustrate the result of the *learning process* for adapting the weights. Close to the line structure, the regularizer *increases* the influence (with larger weights) of neighbors whose distance information matches the prescribed ground truth label. Away from the line structure, the regularizer has learned to *suppress* (with small weights) neighbors that belong to a line structure.



(a) training data **(b) ground truth** **(c) *local rounding* of distance information** **(d) labeling with *optimal* weights**



(e) Optimal weight patches

Figure 5.4: Training phase: Optimal weight patches. TOP ROW: **(a)** Close-up view of training data (10×10 pixel region). **(b)** The corresponding ground truth section. **(c)** Local label assignments. **(d)** Correct labeling using adapted optimal weights. BOTTOM ROW: **(e)** The corresponding optimal weight patches (10×10 grid), one patch for each pixel. Close to the line structure, the regularizer increases the influence of neighbors on the geometric averaging of assignments whose distances match the prescribed ground truth labels. Away from the line structure, the regularizer has learned to *suppress* with small weights neighbors belonging to a line structure.

Test Phase

During the training phase, optimal weights were associated with all training features through optimization, based on ground truth and a corresponding objective function. In the test phase with *novel* data and features, appropriate weights have to be *predicted* because ground truth is no longer available. We realize this by extracting a *coreset* [55] from the output generated by Algorithm 5.1 during the training phase, and constructing a map from novel features to weights, as described next.

Coreset. Let $\Omega^* \in \mathcal{P}$ denote the set of optimal weight patches generated by Algorithm 5.1, and let P^* denote the set of all 15×15 patches of local label assignments based on the corresponding training features and distance (5.23). We partition P^* into three classes: thin *line structures*, *homogeneous* regions and *texture*, and extracted for each class separately 225 prototypical patches by k-means clustering. To each of these patches and the corresponding cluster, a *prototypical weight patch* was assigned, namely the weighted geometric mean of all *optimal* weight patches in Ω^* belonging to that cluster. As weights for the averaging we used the Euclidean distance between the respective patches of local label assignments and the corresponding cluster centroid.

Figure 5.5 depicts 10 pairs of patches of prototypical label assignments and weights, for each of the three classes: line, homogeneous, texture. Comparing these weight patches with the optimal patches depicted by Figure 5.4, we observe that the former are regularized (smoothed) by geometric averaging and, in this sense, summarize and represent all optimal weights computed during the training phase.

Mapping features to weights. For each *novel* test image, we extract features using the same procedure as done in the *training phase* and compute at each pixel i the patch of local label assignments. For the latter patch, the closest patch of local label assignments of the *coreset* is determined, and the corresponding weight patch is assigned to pixel i.

Note that the patch size 15×15 of local label assignments is chosen larger as the patch size 9×9 of the weights that is used during training and testing. The former larger neighborhood defines the local 'feature context' that is used to predict weights for novel data.

Inference (labeling novel data). In the test phase, we use the modified linear assignment flow and all parameter values in the same way, as was done during training. The only difference is that *predicted* weight patches are used for regularization, as described above.

Results. Figure 5.6 shows a result of the test phase. The *top row* shows the input data (panels (a) and (c)), whereas ground truth (b) is only shown for visual comparison. The *bottom row* shows the results obtained using uniform weights (d) and predicted weights (e). The latter result clearly demonstrated the impact of weight *adaptivity*. This aspect is further illustrated in panel (f).

Figure 5.7 shows *predicted* weight patches for novel test data in the same format as Figure 5.4 depicts *optimal* weight patches computed during training. The similarity of predicted and optimal weights for pixels close and away from local line structure, demonstrates that the approach generalizes well to novel data. Since all data are *randomly* generated, the results of Fig. 5.6 and Fig. 5.7 are representative for the entire image class.

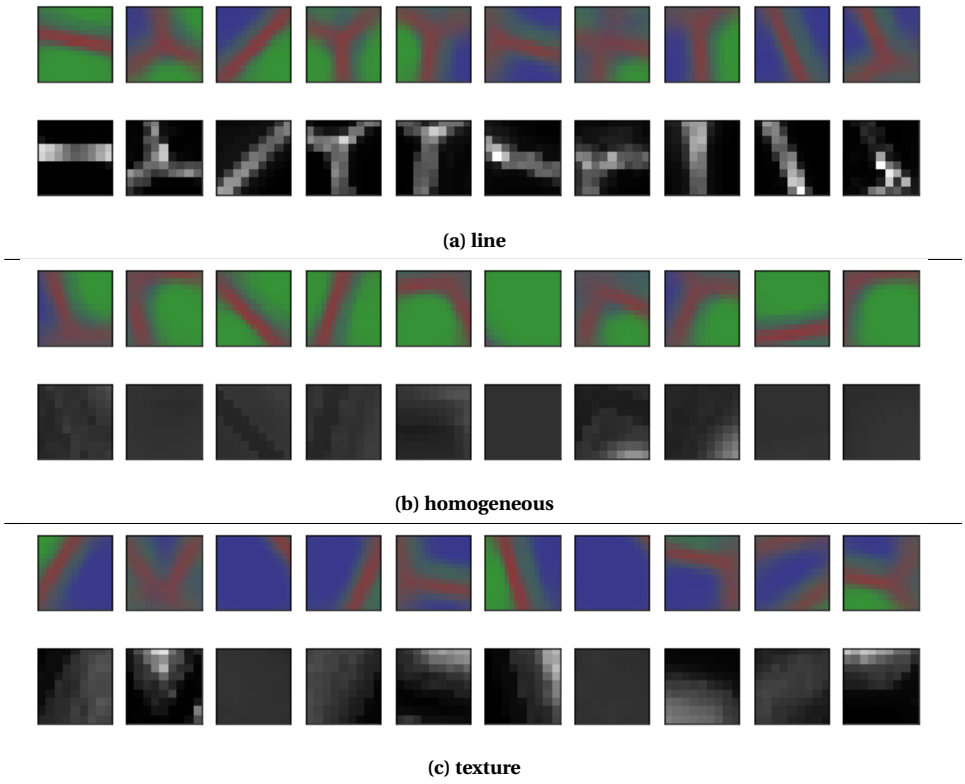
5.4.2 Pattern Formation by Label Transport

In this section we illustrate the model expressiveness of the assignment flow. Specifically, we choose *quite different* labelings as input and target data, respectively, and show that our learning approach can determine weights that 'connect' these patterns by the assignment flow. This shows that the weights which determine the regularization properties of the assignment flow actually encode information for pattern *formation*. Finally, we briefly point out and illustrate limitations of our approach.

Pattern Completion

The *top row* of Figure 5.8 shows *input* and *target* labelings. The *second row* illustrates our approach to weight parameter learning using the *linear* assignment flow: Starting with uniform weights and imposing the very sparse information of the input labeling as constraint, the weights are adapted by the Riemannian gradient flow on the parameter manifold and effectively steer the assignment flow to the target labeling. After convergence we obtain the optimal weights Ω^* and insert them into the original *nonlinear* assignment flow. The evolution corresponding label assignments is shown by the *third row* of Figure 5.8. The fact that the label assignment at the final time T is close to the target labeling which the linear assignment flow reaches exactly, confirms the close approximation of the nonlinear flow by the linear assignment flow, as already demonstrated in [81] in a completely different way.

The rightmost panel of the *top row* shows, for each pixel, the deviation of the optimal weight patch from uniform weights. While it is obvious that the 'source labeling' of the input data receive large weights, the spatial arrangement of weights at all other locations is hard to predict beforehand, by humans. This is why *learning* them is necessary.

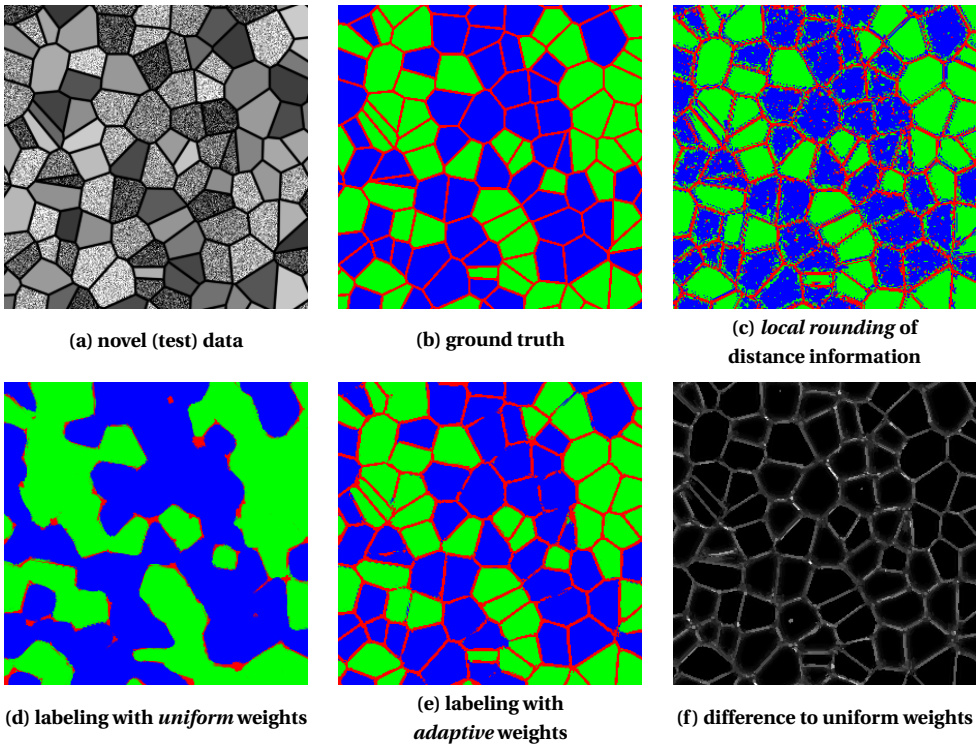


(a) line

(b) homogeneous

(c) texture

Figure 5.5: Coreset visualization. This plot shows 3 × 10 prototypical patches of local label assignments and the corresponding weight patches of the coreset, for each of the 3 classes. **(a)** 10 prototypical pairs of the class *line*. Weight patches 'know' to which neighbors large weights have to be assigned, such that the local line structure is labeled correctly. **(b)** Weight patches of the *homogeneous* label class are almost uniform, which is plausible, because the noisy assignments can be filtered most effectively. **(c)** The weight patches of the *texture* label are comparable to the *homogeneous* ones and almost uniform, for the same reason. (Color code {■, ■, ■} = {line, homogeneous, texture}).



(a) novel (test) data **(b) ground truth** **(c) *local rounding* of distance information**

(d) labeling with *uniform* weights **(e) labeling with *adaptive* weights** **(f) difference to uniform weights**

Figure 5.6: Test phase: Labeling results. TOP ROW: **(a)** Randomly generated novel input data, **(b)** the corresponding ground truth and **(c)** the local label assignments. BOTTOM ROW: **(d)** Labeling using *uniform* weights fails to detect and label line structures. **(e)** *Adaptive* regularizer based on *predicted* weights yields a labeling that largely agrees with the ground truth. Panel **(f)** illustrates weights adaptivity at each pixel in terms of the distance of the *predicted* weight patch to the *uniform* weight.

Transporting and Enlarging Label Assignments

We repeat the experiment of the previous section using the academic scenario illustrated by Figure 5.9. A major difference is that locations of the input labeling *do not* form a subset of the locations of the target labeling. As a consequence, the corresponding 'mass' of assignments has to be both *transported and enlarged*.

The results shown by Figure 5.9 are quite similar to those of Figure 5.8, such that the corresponding comments apply likewise. Looking at the optimal weight patches in terms of their deviation from uniform weights (rightmost panel of Fig. 5.9, *top row*)

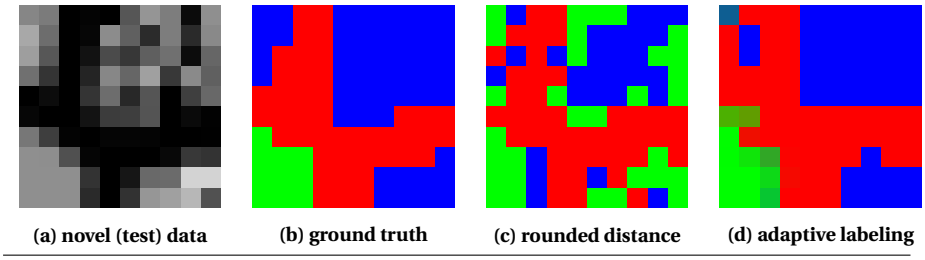
it is both interesting and not too difficult to understand – after convergence and informally by visual inspection – how these weights encode this particular 'label transport'. However, predicting these weights and certifying their optimality *beforehand*, seems to be an infeasible task. For example, it is hard to predict that the creation of intermediate locations where assignment mass temporarily accumulates (clearly visible in Fig. 5.9), effectively optimizes the constrained functional (5.1). *Learning* these weights, on the other hand, just requires to apply our approach.

Parameter Learning vs. Optimal Control

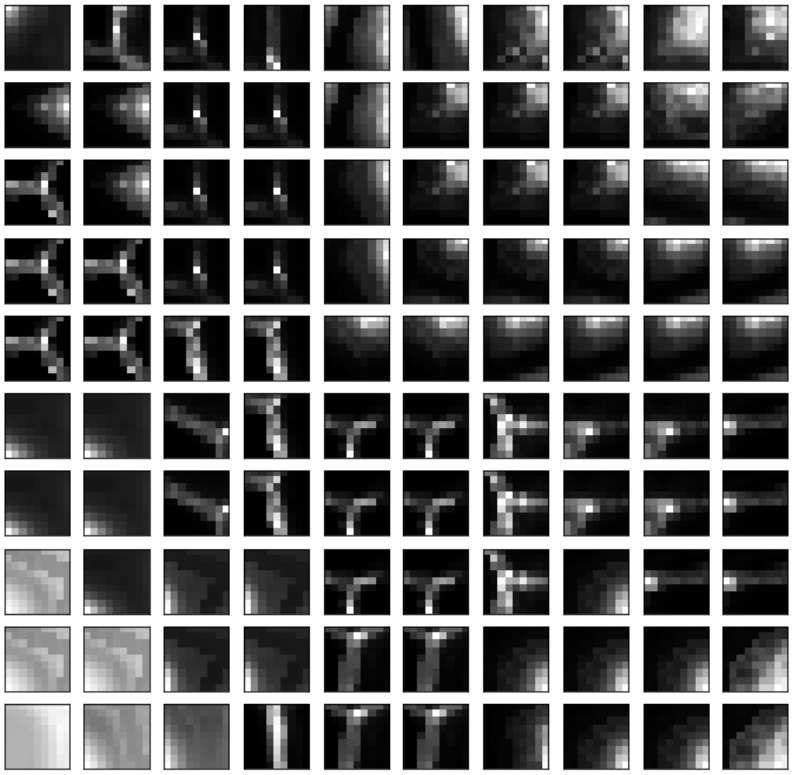
The limitations of our parameter learning approach are illustrated by Fig. 5.10. In this experiment, we simply *exceed* the time horizon in order to inspect labelings induced by the linear assignment flow *after* the point of time T, that was used for determining optimal weights in the training phase. Starting with T, Figure 5.10 shows these labelings for both experiments corresponding to Figures 5.8 and 5.9.

Unlike the fern pattern (*top row*) where the initial label locations form a subset of the target locations and are imposed as constraints, the 'moving mass pattern' (*bottom row*) is *unsteady* in the following quite natural sense: the linear assignment flow simply continues transporting mass beyond time T. As a result, assignments to the white label are transported to locations of the black target pattern. Hence, the target pattern is first created up to time T and destroyed afterwards.

This behavior is not really a limitation, but a consequence of merely learning *constant* weight parameters. Due to the formulation of the optimization problem (5.1), optimal weights not only encode the 'knowledge' how to steer the assignment flow in order to solve the problem, but also the *time period* after which the task has to be completed. Fixing this issue requires a higher-level of adaptivity: weight *functions* depending on time and the current state of assignments would have to be estimated, that may be adjusted online through feedback in order to *control* the assignment flow in a more flexible way.



(a) novel (test) data **(b) ground truth** **(c) rounded distance** **(d) adaptive labeling**



(e) predicted weight patches

Figure 5.7: Test phase: Predicted weight patches. TOP ROW: **(a)** Close-up view of *novel* data (10 × 10 pixel window). **(b)** Corresponding ground truth section (just for visual comparison, *not* used in the experiment). **(c)** Local label assignment. **(d)** Labeling result using adaptive regularization with predicted weights. BOTTOM: **(e)** Corresponding predicted weight patches (10 × 10 grid), one patch for each pixel of the test data (a). The *predicted* weight patches behave similar to the *optimal* weight patches depicted by Fig. 5.4, that were computed during the training phase (for different data). This shows that our approach generalizes to novel data.

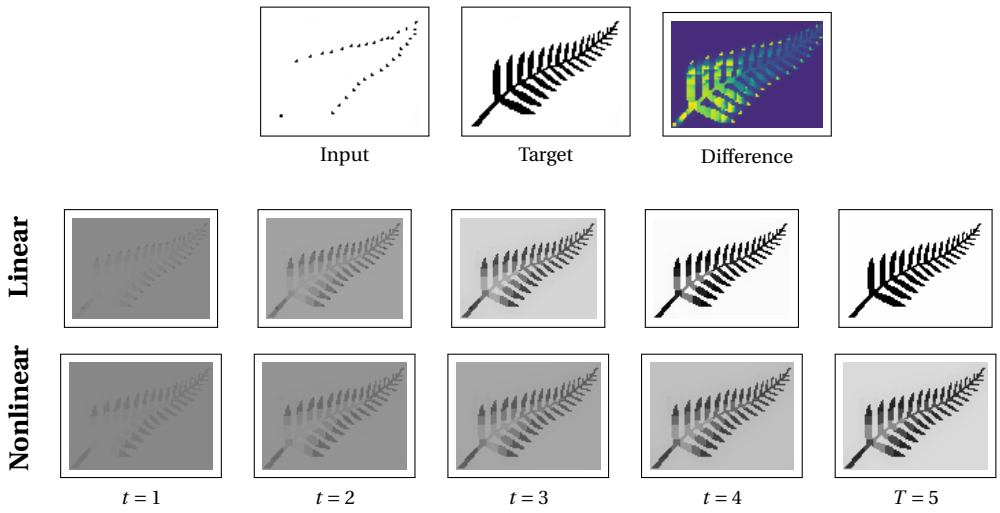


Figure 5.8: Pattern completion. This figure illustrates the model expressiveness of the assignment flow. TOP ROW: Input and target labelings. The task is to estimate weights in order to steer the assignment flow to the target labeling. The rightmost panel illustrates, for each pixel, the distance of the *optimal* weight patch from *uniform* weights. MIDDLE ROW: Label assignments of the linear assignment flow during weight parameter estimation. The Riemannian gradient flow on the parameter manifold effectively steers the flow to the target labeling. BOTTOM ROW: Label assignments of the *nonlinear* assignment flow using the optimal weights that were estimated using the *linear* assignment flow. Closeness of both labeling patterns at the final point of time $T = 5$ demonstrates that the linear assignment flow provides a good approximation of the full nonlinear flow.

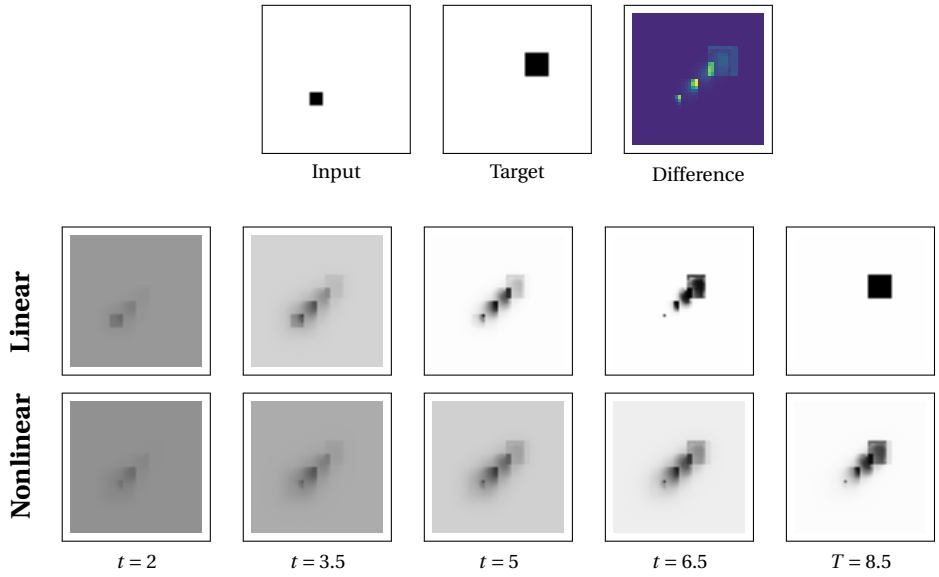


Figure 5.9: Transporting and enlarging label assignments. We use the same set-up as in Fig. 5.8. TOP ROW: Label locations of the input data *do not* form a subset of the target locations. Thus, 'mass' of label assignments has to be both *transported* and *enlarged*. Rightmost panel: Distance of the *optimal* weight patch from uniform weights, for every pixel. MIDDLE ROW: Applying our approach to (5.1) effectively solves the problem. BOTTOM ROW: Inserting the optimal weights that are computed using the *linear* assignment flow, into the *nonlinear* assignment flow, gives a similar result and underlines the good approximation property of the linear assignment flow. It is interesting to obverse that computing the Riemannian gradient flow on the parameter manifold entails 'intermediate locations' where assignment mass accumulates temporarily. This underlines the necessity of learning, since it seems hard to predict such an *optimal* regularization strategy beforehand.

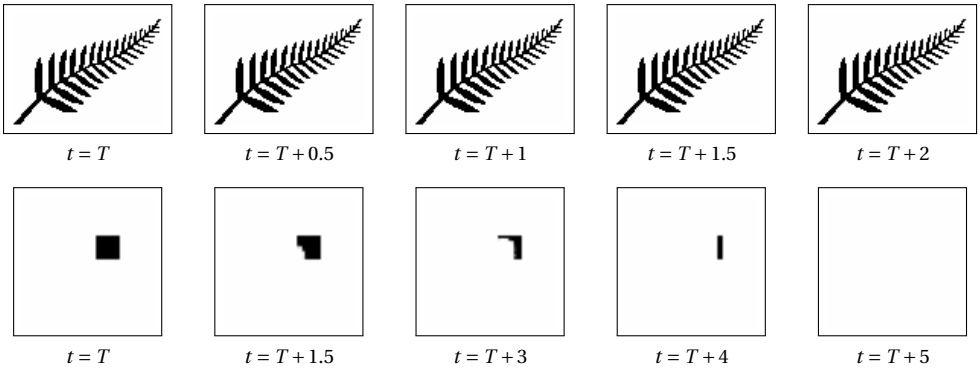


Figure 5.10: Parameter learning vs. optimal control. The plots show label assignments by computing the assignment flow *beyond* the final point of time T used during training, for the experiments corresponding to Figures 5.8 and 5.9. Unlike the pattern completion experiment (*top row*) where few locations of initial label assignments are imposed as constraint, the target pattern (bottom row, at time T) of the moving-mass experiment is *unsteady* in the following sense: at time T, the flow continues to transport mass which eventually erases the target pattern with assignments of the white background label. The reason is that *constant* parameters are only learned that not only encode the 'knowledge' how to steer the flow to the target pattern but also the time period $[0, T]$ for accomplishing this task. In order to overcome this limitation, weight *functions* depending on time and the current state of the assignments would have to be estimated by applying techniques of optimal control.

Chapter 6

Conclusion

Summary. With the present work we hope to have contributed to the rich literature on algorithms for graphical models. Hopefully, the interplay of different research areas (image processing, differential geometry, optimal control theory and machine learning) and the corresponding different perspectives have provided new insights into the recent advances in hierarchical architectures. Especially, we think it is worth to investigate deep networks whose parametrizations and internal representations are not fully understood from the perspective of optimal control theory. We hope that this thesis and possible extensions of it further the understanding of these architectures.

In Chapter 3 we have presented the work of [8] in detail as it serves as the basis for our main contributions presented in Chapter 4 and 5.

Based on this smooth geometric setting, we presented in Chapter 4 a novel *MAP inference* approach to the evaluation of discrete graphical models. The coupling measures between adjacent nodes are incorporated by regularized Wasserstein distances. The novel algorithm propagates in parallel the corresponding local gradients – called *Wasserstein messages* – along edges. These messages are lifted to the assignment manifold and drive a Riemannian gradient flow that terminates at an integral labeling. In contrast to established belief propagation, the local marginalization constraints are satisfied throughout the process. A single parameter facilitates the trade-off between accuracy of optimization and speed of convergence.

Conversely, in Chapter 5 we introduced a *parameter learning* approach for image labeling based on the assignment flow. In a supervised setting, we estimated weights for geometric averaging of label assignments in order to steer the flow to prescribed ground truth labelings. By using a class of symplectic partitioned Runge–Kutta methods, we have shown that this task can be accomplished by numerically integrating the adjoint system in a consistent way. Consistent means that discretization and differentiation for the computation of the adjoint sensitivity *commute*. An additional convenient property of our approach is that the parameter manifold has the mathematical structure of an assignment manifold, such that Riemannian gradient descent can be used for efficiently solving the training problem.

The output of the training phase served as a core set that consists of features extracted from training images and the respective optimal weights. In order to label novel data, a mapping has to be specified that predicts optimal weights for this unseen data. We solved this task by nearest-neighbor prediction after partitioning the core set using k-means clustering and geometric averaging of the weights, separately for each cluster. We evaluated this approach for a class of images involving line structures where just using uniform weights inevitably fails. We additionally conducted experiments that highlight the model expressiveness of the assignment flow and also limitations caused by merely learning *constant* parameters.

Future Work. The inference approach presented in Chapter 4 motivates the following research directions:

- *Generalizing our approach to tighter relaxations.* Investigate if our approach can be generalized to relaxations based on hypergraphs and corresponding entropy approximations [80, 51].
- *Designing more advanced numerical schemes.* A possible direction is to advance the numerical schemes by using multiple spatial scales.
- *Addressing applications using graphical models with higher edge connectivity.* Since established inference algorithms based on convex programming noticeably slow down with increasing edge connectivity, it is worth investigating how our approach scales for these applications.

The main insights of Chapter 5 include the following; Regarding numerical optimization for parameter learning in connection with image labeling, our approach is more satisfying than working with discrete graphical models, where parameter learning requires evaluating the partition function, which is a much more involved task when working with cyclic grid graphs. This latter problem of computational *statistics* shows up in our scenario in similar form as the problem of designing the prediction map from features to weight parameters. An essential difference between these two scenarios is that by restricting the scope to statistical predictions at a *local* scale, i.e. only within small windows, the prediction task becomes *manageable*, since, regarding numerical optimization, no further approximations are involved at all.

Nevertheless, the approach is by no means finished and should be seen as a first step towards more advanced and expressive architectures. In particular, the approach could be extended in the following promising research directions:

- *Building a parametrized prediction map into the learning framework.* The idea is to design a parametrized prediction map that is learned during the learning process. This would cast our two-step approach into a single process, commonly called *end-to-end learning*. This would have the benefit that the number of parameters can be reduced to a minimal but sufficient parametrization. However, this could make the learning task much more involved depending on the given image data.

- *Learning weight functions instead of constant parameters.* In order to *control* the assignment flow in a more flexible way and to reach a higher-level of adaptivity, weight functions depending on time and the current state of assignments would have to be estimated and adjusted online through feedback.

- *Composing several assignment flows in a hierarchical fashion.* Intuitively, this results in components which evolve on different time scales. The design of such complex flows, their proper numerical integration over all scales and a good understanding of suitable data representations at intermediate scales define challenging research tasks.

Appendix A

Proofs

A.1 Proofs of Chapter 2

Proofs of Section 2.3.1

Proof (Theorem 2.4) A detailed proof can be found in [32, Chapter I.14, Theorem 14.1]. In order to make this chapter self-contained, a sketch follows.

The integral representation of the solution to (2.51c) is given by $x(t,p) = x_0 + \int_0^t f(x(s),p,s)ds$. Differentiating with respect to p and exchanging integration and differentiation by the theorem of Lebesgue yields

$$d_p x(t,p) = d_p x_0 + \int_0^t d_p f(x(s),p,s)ds \tag{A.1a}$$

$$= d_p x_0 + \int_0^t d_x f(x(s),p,s)d_p x(s,p) + d_p f(x(s),p,s)ds. \tag{A.1b}$$

Substituting $\delta(t) = d_p x(t,p)$, gives

$$\delta(t) = \delta_0 + \int_0^t d_x f(x(s),p,s)\delta(s) + d_p f(x(s),p,s)ds, \tag{A.2}$$

which is the integral representation of the trajectory $\delta(t)$ solving (2.55). □

Proofs of Section 2.3.3

Proof (Theorem 2.9) A proof can be found, e.g., in [17]. However, in order to make this thesis self-contained, we include a proof here. Setting up the Lagrangian

$$\mathcal{L}(x,p,\lambda) = \mathcal{C}(x(T)) - \int_0^T \langle \lambda, F(\dot{x},x,p,t)\rangle dt \tag{A.3}$$

with multiplier $\lambda(t)$ and $F(\dot{x}, x, p, t) := \dot{x} - f(x, p, t) \equiv 0$, we get with $\Phi(p) = \mathcal{C}(x(T))$ from (2.52)

$$\nabla\Phi = \nabla_p \mathcal{L} = d_p x(T)^\top \nabla\mathcal{C}\big(x(T)\big) - \int_0^T \Big(d_{\dot{x}} F d_p \dot{x} + d_x F d_p x + d_p F\Big)^\top \lambda \, dt, \tag{A.4}$$

where integration applies componentwise. By using $d_{\dot{x}} F = I$, where I denotes the identity matrix, we partially integrate the first term under the integral,

$$\int_0^T d_p \dot{x}^\top \lambda \, dt = d_p x^\top \lambda \Big|_{t=0}^T - \int_0^T d_p x^\top \dot{\lambda} \, dt. \tag{A.5}$$

We further obtain with $d_p F = -d_p f$ and $d_x F = -d_x f$

$$\nabla\Phi = d_p x(T)^\top \nabla\mathcal{C}(x(T)) - d_p x^\top \lambda \Big|_{t=0}^T + \int_0^T d_p x^\top \dot{\lambda} \, dt + \int_0^T \Big(d_x f d_p x + d_p f\Big)^\top \lambda \, dt \tag{A.6a}$$

$$\begin{aligned} &= d_p x(T)^\top \nabla\mathcal{C}(x(T)) - d_p x(T)^\top \lambda(T) + d_p x(0)^\top \lambda(0) \\ &\quad + \int_0^T d_p x^\top \dot{\lambda} + d_p x^\top d_x f^\top \lambda + d_p f^\top \lambda \, dt. \end{aligned} \tag{A.6b}$$

We consider systems where the initial value x_0 is independent of the parameter p, i.e. $d_p x(0) = 0$. Additionally factoring out the unknown Jacobian $d_p x$, we obtain

$$= d_p x(T)^\top \Big(\nabla\mathcal{C}(x(T)) - \lambda(T)\Big) + \int_0^T d_p x^\top \Big(\dot{\lambda} + d_x f^\top \lambda\Big) + d_p f^\top \lambda \, dt. \tag{A.6c}$$

Now, by choosing $\lambda(t)$ such that conditions (2.67b) are fulfilled, i.e.

$$\dot{\lambda}(t) = -d_x f^\top \lambda(t), \qquad \lambda(T) = \nabla_x \mathcal{C}(x(T)),$$

we finally obtain

$$\nabla\Phi = \int_0^T d_p f^\top \lambda(t) \, dt. \tag{A.7}$$

□

Proof (Lemma 2.11) Since we evaluate all occurring functions and their derivatives at the same points p_0, γ_0 and λ_0, we drop them as arguments in the following, to simplify notation.
(i) Equation (2.76a) directly follows by differentiating $\mathcal{L}$ with respect to λ at $(p_0, \gamma_0, \lambda_0)$.

(ii) Equation (2.76b) is immediately obtained by differentiating $\mathcal{L}$ with respect to γ at $(p_0, \gamma_0, \lambda_0)$. Since $d_\gamma \phi$ is invertible at (p_0, γ_0), the resulting linear system uniquely determines the vector λ_0.
(iii) Next, we show that this λ_0 also satisfies the first equation (2.75). By differentiating $\phi(p, \gamma) = 0$ with respect to p at (p_0, γ_0), we obtain

$$d_\gamma \phi d_p \gamma_0 + d_p \phi = 0 \quad \overset{d_\gamma \phi \text{ is invertible}}{\iff} \quad d_p \gamma_0 = -(d_\gamma \phi)^{-1} d_p \phi. \tag{A.8}$$

We will make use of this identity for $d_p \gamma_0$ in the following. Differentiating Φ with respect to p at p_0 and by the chain rule, we obtain

$$\begin{aligned} \nabla \Phi &= \nabla_p \mathcal{C} + d_p \gamma_0^\top \nabla_\gamma \mathcal{C} \overset{(2.76b)}{=} \nabla_p \mathcal{C} - d_p \gamma_0^\top d_\gamma \phi^\top \lambda_0 && \text{(A.9a)} \\ &\overset{(A.8)}{=} \nabla_p \mathcal{C} + \left((d_\gamma \phi)^{-1} d_p \phi\right)^\top d_\gamma \phi^\top \lambda_0 = \nabla_p \mathcal{C} + d_p \phi^\top \lambda_0 && \text{(A.9b)} \\ &= \nabla \mathcal{L}, && \text{(A.9c)} \end{aligned}$$

which shows (2.75). □

Proof (Theorem 2.10) We begin by stating the Lagrangian of problem (2.70)

$$\begin{aligned} \mathcal{L}(x, p, \lambda) = \mathcal{C}\big(x_N\big) - \lambda_0^\top (x_0 - x(0)) - \sum_{n=0}^{N-1} \lambda_{n+1}^\top \Big[x_{n+1} - x_n - h_n \sum_{i=1}^{s} b_i k_{n,i} \Big] \\ - \sum_{n=0}^{N-1} h_n \sum_{i=1}^{s} b_i \Lambda_{n,i}^\top \Big[k_{n,i} - f(X_{n,i}, p, t_n + c_i h_n) \Big]. \end{aligned} \tag{A.10}$$

In order to apply Lemma 2.11, we explain which role the variables γ, λ, ϕ play in this situation:

1. *Intermediate stages:* The vector γ represents all intermediate stages related to the evaluation of the function $\Phi(p) = \mathcal{C}(x_N(p))$, i.e. all intermediate values x_i and stages k_i of the Runge–Kutta method. These variables are stacked and arranged as follows

$$\gamma = \begin{bmatrix} x_0 \\ \gamma_0 \\ \gamma_1 \\ \vdots \\ \gamma_{N-1} \end{bmatrix} \in \mathbb{R}^{d'}, \quad \text{with} \quad \gamma_n = \begin{bmatrix} k_n \\ x_{n+1} \end{bmatrix} \in \mathbb{R}^{(s+1)n_x}, \quad \text{and} \quad k_n = \begin{bmatrix} k_{n,1} \\ \vdots \\ k_{n,s} \end{bmatrix} \in \mathbb{R}^{s n_x}. \tag{A.11}$$

2. *Lagrange multiplier:* The vector λ contains all Lagrange multipliers in (A.10) belonging to the constraints (2.70b)-(2.70d). The multipliers are stacked and arranged as follows

$$\lambda = \begin{bmatrix} -\lambda_0 \\ -\Lambda_0 \\ \vdots \\ -\lambda_{N-1} \\ -\Lambda_{N-1} \\ -\lambda_N \end{bmatrix} \in \mathbb{R}^{d'}, \quad \text{with} \quad \Lambda_n = \begin{bmatrix} h_n b_1 \Lambda_{n,1} \\ \vdots \\ h_n b_s \Lambda_{n,s} \end{bmatrix} \in \mathbb{R}^{s n_x}. \tag{A.12}$$

3. *Intermediate mappings:* Analogously, the vector ϕ contains all intermediate mappings ϕ_n, for $n = 1, \dots, N-1$ of the computation of $\Phi(p) = C(x_N(p))$. In our situation, ϕ is the concatenation of the *forward* Runge–Kutta evaluation, which we express using the Kronecker-product as

$$\phi = \begin{bmatrix} x_0 - x(0) \\ \Psi_1 \\ \Psi_2 \\ \vdots \\ \Psi_{N-1} \end{bmatrix} \in \mathbb{R}^{d'}, \quad \text{with} \quad \Psi_n = \begin{bmatrix} k_n - F_n(X_n, p) \\ x_{n+1} - x_n - h_n (b^\top \otimes I_{n_x}) k_n \end{bmatrix} = \begin{bmatrix} \Psi_{n,1} \\ \Psi_{n,2} \end{bmatrix} \in \mathbb{R}^{(s+1) n_x}, \tag{A.13}$$

where $\Psi_{n,1} \in \mathbb{R}^{s n_x}$ and $\Psi_{n,2} \in \mathbb{R}^{n_x}$, as well as

$$F_n(X_n, p) = \begin{bmatrix} f(X_{n,1}, p, t_n + c_1 h_n) \\ \vdots \\ f(X_{n,s}, p, t_n + c_s h_n) \end{bmatrix}, \qquad X_n = \mathbb{1}_s \otimes x_n + h_n (A \otimes I_{n_x}) k_n = \begin{bmatrix} X_{n,1} \\ \vdots \\ X_{n,s} \end{bmatrix}. \tag{A.14}$$

We proceed by computing the Jacobian $d_\gamma \phi$. Note that the intermediate variables γ_n (A.11) are only contained in the intermediate mappings Ψ_n (A.13), which results in a sparse block structure of the overall Jacobian $d_\gamma \phi$.

1. *Small block matrices:* Each small block matrix represents the derivative of the n-th iteration step Ψ_n and is given by

$$d_{(x_n,k_n,x_{n+1})}\Psi_n = \begin{bmatrix} d_{x_n}\Psi_{n,1} & d_{k_n}\Psi_{n,1} & d_{x_{n+1}}\Psi_{n,1} \\ d_{x_n}\Psi_{n,2} & d_{k_n}\Psi_{n,2} & d_{x_{n+1}}\Psi_{n,2} \end{bmatrix} = \begin{bmatrix} D_n & A_n & \\ -I_{n_x} & B_n^\top & I_{n_x} \end{bmatrix}, \tag{A.15}$$

with

$$A_n = I_{sn_x} - h_n d_x F_n(X_n, p)(A \otimes I_{n_x}), \tag{A.16a}$$
$$B_n^\top = -h_n b^\top \otimes I_{n_x}, \tag{A.16b}$$
$$D_n = -d_x F_n(X_n, p)(\mathbb{1}_s \otimes I_{n_x}), \tag{A.16c}$$

where A and b are the Runge-Kutta coefficients given by the left tableau of Fig. 2.1.

2. *Sparse block structure:* The overall Jacobian $d_\gamma \phi$ consists of $N-1$ blocks (one for each iteration) of the form (A.15) and is given by

$$d_\gamma \phi = \begin{bmatrix} I_{n_x} & & & & & & & \\ D_1 & A_1 & & & & & & \\ -I_{n_x} & B_1^\top & I_{n_x} & & & & & \\ & & D_2 & A_2 & & & & \\ & & -I_{n_x} & B_2^\top & I_{n_x} & & & \\ & & & \ddots & \ddots & \ddots & & \\ & & & & & D_{N-1} & A_{N-1} & \\ & & & & & -I_{n_x} & B_{N-1}^\top & I_{n_x} \end{bmatrix}. \tag{A.17}$$

3. *Invertibility of $d_\gamma \phi$:* A matrix M is invertible if $\det M \neq 0$. Since the matrix $d_\gamma \phi$ is lower block diagonal, its determinant is given by

$$\det d_\gamma \phi = \det A_1 \cdot \ldots \cdot \det A_{N-1} \tag{A.18}$$

Thus, we only need to show that $\det A_n \neq 0$ for all $n = 1, \ldots, N-1$. Equation (A.16a) reads in a more compact form

$$A_n = (I_{sn_x} - h_n M), \qquad \text{with } M := d_x F_n(X_n, p)(A \otimes I_{n_x}). \tag{A.19}$$

We show $\det A_n \neq 0$ by using the equivalent statement $\ker(A_n) = \{0\}$. Now, let $x \in \mathbb{R}^{sn_x} \setminus \{0\}$, then

$$\|x\| \leq \|x - h_n M x\| + \|h_n M x\| \leq \|A_n x\| + \|h_n M\| \|x\|. \tag{A.20}$$

By using the row-sum norm $\|\cdot\|_\infty$, we have

$$\begin{aligned} \|h_n M\|_\infty &\overset{(A.19)}{=} h_n \|d_x F_n(X_n, p)(A \otimes I_{n_x})\|_\infty \leq h_n \|d_x F_n(X_n, p)\|_\infty \|(A \otimes I_{n_x})\|_\infty \\ &< h_n L \max_{i=1,\dots,s} \sum_{j=1}^{s} |a_{ij}| \overset{(2.60)}{<} 1, \end{aligned} \tag{A.21}$$

where L denotes the Lipschitz constant of f and the step-size h_n satisfies the assumption (2.60). Substituting (A.21) into (A.20) gives

$$\|x\| < \|A_n x\| + \|x\| \iff 0 < \|A_n x\| \iff x \notin \ker(A_n). \tag{A.22}$$

Since, the kernel of A_n is trivial, A_n is invertible and consequently the overall Jacobian $d_\gamma \phi$ as well.

Now we are in a position to apply Lemma 2.11. More precisely, (2.76b) tells us that the vector λ is uniquely determined by the linear system $d_\gamma \phi^\top \lambda = -\nabla_\gamma \mathcal{C}$. In our situation, this system is given by

$$\begin{bmatrix} I_{n_x} & D_1^\top & -I_{n_x} & & & & & & \\ & A_1^\top & B_1 & & & & & & \\ & & I_{n_x} & D_2^\top & -I_{n_x} & & & & \\ & & & A_2^\top & B_2 & \ddots & & & \\ & & & & I_{n_x} & \ddots & & & \\ & & & & & \ddots & D_{N-1}^\top & -I_{n_x} & \\ & & & & & & A_{N-1}^\top & B_{N-1} & \\ & & & & & & & I_{n_x} & \end{bmatrix} \begin{bmatrix} -\lambda_0 \\ -\Lambda_0 \\ -\lambda_1 \\ -\Lambda_1 \\ \vdots \\ -\lambda_{N-1} \\ -\Lambda_{N-1} \\ -\lambda_N \end{bmatrix} = - \begin{bmatrix} 0 \\ 0 \\ 0 \\ 0 \\ \vdots \\ 0 \\ 0 \\ \nabla_x \mathcal{C}(x_N) \end{bmatrix}. \tag{A.23}$$

We determine the exact identity of λ by backward substitution:

1. From the last row of (A.23), we immediately obtain

$$\lambda_N = \nabla_x \mathcal{C}(x_N). \tag{A.24}$$

2. Next, we prove equation (2.72a). For each $n = 0, \dots, N-1$, we obtain

$$0 = \begin{bmatrix} I_{n_x} & D_n^\top & -I_{n_x} \end{bmatrix} \begin{bmatrix} \lambda_n \\ \Lambda_n \\ \lambda_{n+1} \end{bmatrix} = \lambda_n + D_n^\top \Lambda_n - \lambda_{n+1} \tag{A.25a}$$

$$= \lambda_n - (d_x F_n(X_n, p)(\mathbb{1}_s \otimes I_{n_x}))^\top \begin{bmatrix} h_n b_1 \Lambda_{n,1} \\ \vdots \\ h_n b_s \Lambda_{n,s} \end{bmatrix} - \lambda_{n+1} \tag{A.25b}$$

$$= \lambda_n - (\mathbb{1}_s^\top \otimes I_{n_x}) d_x F_n(X_n, p)^\top \begin{bmatrix} h_n b_1 \Lambda_{n,1} \\ \vdots \\ h_n b_s \Lambda_{n,s} \end{bmatrix} - \lambda_{n+1} \tag{A.25c}$$

$$= \lambda_n - h_n \sum_{i=1}^{s} b_i d_x f(X_{n,i}, p, t_n + c_i h_n)^\top \Lambda_{n,i} - \lambda_{n+1}. \tag{A.25d}$$

$$\lambda_{n+1} = \lambda_n + h_n \sum_{i=1}^{s} b_i \ell_{n,i}, \quad \text{with} \quad \ell_{n,i} = -d_x f(X_{n,i}, p, t_n + c_i h_n)^\top \Lambda_{n,i}. \tag{A.25e}$$

3. The last equation (2.72c) follows by

$$0 = \begin{bmatrix} 0 & A_n^\top & B_n \end{bmatrix} \begin{bmatrix} \lambda_n \\ \Lambda_n \\ \lambda_{n+1} \end{bmatrix} \tag{A.26a}$$

$$= A_n^\top \Lambda_n + B_n \lambda_{n+1} \tag{A.26b}$$

$$= \left(I_{sn_x} - h_n d_x F_n(X_n, p)(A \otimes I_{n_x}) \right)^\top \Lambda_n - h_n (b \otimes I_{n_x}) \lambda_{n+1} \tag{A.26c}$$

$$= \left(I_{sn_x} - h_n (A^\top \otimes I_{n_x}) d_x F_n(X_n, p)^\top \right) \Lambda_n - h_n (b \otimes I_{n_x}) \lambda_{n+1}. \tag{A.26d}$$

In the following, we consider the i-th entry of the previous equation, i.e. $h_n b_i \Lambda_{n,i}$ of Λ_n with $i = 1, \dots, s$.

$$0 \quad = \quad h_n b_i \Lambda_{n,i} - h_n^2 \sum_{j=1}^{s} a_{ji} b_j \partial_x f(X_{n,j}, p, t_n + c_j h_n)^\top \Lambda_{n,j} - h_n b_i \lambda_{n+1} \tag{A.27a}$$

$$\Lambda_{n,i} = \lambda_{n+1} + h_n \sum_{j=1}^{s} \frac{a_{ji} b_j}{b_i} d_x f(X_{n,j}, p, t_n + c_j h_n)^\top \Lambda_{n,j} \tag{A.27b}$$

$$\overset{(A.25e)}{=} \lambda_n + h_n \sum_{i=1}^{s} b_i \ell_{n,i} - h_n \sum_{j=1}^{s} \frac{a_{ji} b_j}{b_i} \ell_{n,j} \tag{A.27c}$$

$$= \lambda_n + h_n \sum_{j=1}^{s} \left(b_j - \frac{a_{ji} b_j}{b_i} \right) \ell_{n,j}, \tag{A.27d}$$

with $\ell_{n,j} = -d_x f(X_{n,j}, p, t_n + c_j h_n)^\top \Lambda_{n,j}$ in (A.27c) and (A.27d).

Finally, we show the formula of the gradient (2.71), which is given by (2.75)

$$\nabla \Phi = \nabla_p \mathcal{C} + d_p \phi^\top \lambda_0 \overset{\nabla_p \mathcal{C} = 0}{=} d_p \phi^\top \lambda_0. \tag{A.28}$$

The Jacobian $d_p \phi^\top$ consists of the following building blocks: For the n-th iteration step Ψ_n the *local* Jacobian with respect to parameter p reads

$$d_p \Psi_n = \begin{bmatrix} d_p \Psi_{n,1} \\ d_p \Psi_{n,2} \end{bmatrix} = \begin{bmatrix} \bar{D}_n \\ 0 \end{bmatrix}, \quad \text{with} \quad \bar{D}_n = -d_p F_n(X_n, p)(\mathbb{1}_s \otimes I_{n_p}). \tag{A.29}$$

By concatenating $N-1$ of these blocks (one for each iteration $n = 1, \dots, N-1$) of (A.29), the overall Jacobian is given by

$$d_p \phi^\top = \begin{bmatrix} 0 & \bar{D}_0^\top & 0 & \bar{D}_1^\top & 0 & \dots & 0 & \bar{D}_{N-1}^\top & 0 \end{bmatrix}. \tag{A.30}$$

Now, formula (2.71) is explicitly given by

$$\nabla \Phi = d_p \phi^\top \lambda_0 \tag{A.31a}$$

$$= \begin{bmatrix} 0 & \bar{D}_0^\top & 0 & \dots & 0 & \bar{D}_{N-1}^\top & 0 \end{bmatrix} \begin{bmatrix} -\lambda_0 \\ -\Lambda_0 \\ -\lambda_1 \\ \vdots \\ -\lambda_{N-1} \\ -\Lambda_{N-1} \\ -\lambda_N \end{bmatrix} = -\sum_{n=0}^{N-1} \bar{D}_n^\top \Lambda_n \tag{A.31b}$$

$$= \sum_{n=0}^{N-1} (d_p F_n(X_n, p)(\mathbb{1}_s \otimes I_{n_p}))^\top \Lambda_n = \sum_{n=0}^{N-1} ((\mathbb{1}_s \otimes I_{n_p})) d_p F_n(X_n, p)^\top \Lambda_n \tag{A.31c}$$

$$\overset{(A.12)}{=} \sum_{n=0}^{N-1} h_n \sum_{i=1}^{s} b_i d_p f(X_{n,i}, p, t_n + c_i h_n)^\top \Lambda_{n,i}. \tag{A.31d}$$

□

A.2 Proofs of Chapter 3

Proofs of Section 3.1.1

Proof (Lemma 3.2) The first property follows by a direct calculation

$$\begin{aligned}\exp_p(x+y) &= \frac{pe^{x+y}}{\langle p, e^{x+y}\rangle} = \frac{\left(pe^x\right)e^y}{\left\langle \left(pe^x\right), e^y\right\rangle} = \frac{\left(\frac{pe^x}{\langle p,e^x\rangle}\right)e^y}{\left\langle \left(\frac{pe^x}{\langle p,e^x\rangle}\right), e^y\right\rangle} \\ &= \frac{\exp_p(x)e^y}{\left\langle \exp_p(x), e^y\right\rangle} = \exp_{\exp_p(x)}(y).\end{aligned}$$

Next we show the second property. Let $\gamma\colon (-\epsilon,\epsilon) \to \mathcal{S}$ be a smooth curve with $\gamma(0) = p \in \mathcal{S}, \dot\gamma(0) = v \in T$. Then, the differential reads

$$\begin{aligned} d_q \exp_p(q)[v] &= \frac{d}{dt}\frac{pe^{\gamma(t)}}{\langle p, e^{\gamma(t)}\rangle}\bigg|_{t=0} \\ &= \frac{\dot\gamma(0)\cdot pe^{\gamma(0)}\cdot\langle p, e^{\gamma(0)}\rangle - pe^{\gamma(0)}\cdot\langle p, \dot\gamma(0)\cdot e^{\gamma(0)}\rangle}{\langle p, e^{\gamma(0)}\rangle^2} \\ &= \frac{v\cdot pe^q\cdot\langle p, e^q\rangle - pe^q\cdot\langle p, v\cdot e^q\rangle}{\langle p, e^q\rangle^2} \\ &= v\cdot\frac{pe^q}{\langle p, e^q\rangle} - \frac{pe^q}{\langle p, e^q\rangle}\cdot\langle v, \frac{pe^q}{\langle p, e^q\rangle}\rangle \\ &= \left(\operatorname{diag}\left(\exp_p(q)\right) - \exp_p(q)\exp_p(q)^T\right)v = R_{\exp_p(q)}[v]\end{aligned}$$

Again, let $\gamma\colon (-\epsilon,\epsilon) \to \mathcal{S}$ be a smooth curve with $\gamma(0) = p \in \mathcal{S}, \dot\gamma(0) = v \in T$. Then, the last property follows by

$$d\exp_p^{-1}(q)[x] = \frac{d}{dt}\Pi_{T_n}\log\frac{q}{\gamma(t)}\bigg|_{t=0} = \Pi_{T_n}\frac{\gamma(0)}{q}\frac{\dot\gamma(0)}{\gamma(0)} = \Pi_{T_n}\frac{v}{q}.$$

□

Proof (Lemma 3.6) By using $\operatorname{Exp}_p^{e,-1}$, the optimality condition (3.16b) reads

$$0 = \sum_{i\in[N]}\omega_i \operatorname{Exp}_p^{e,-1}(p_i) \overset{(3.12b)}{=} \sum_{i\in[N]}\omega_i R_p \log\frac{p_i}{p} \tag{A.32a}$$

$$= R_p\Big[\sum_{i\in[N]} \omega_i \log p_i\Big] - R_p\Big[\log p\Big]. \tag{A.32b}$$

According to Remark 3.1, with $R_p = R_p\Pi_{T_n} = \Pi_{T_n} R_p$, we have

$$= \Pi_{T_n} R_p\Big[\sum_{i\in[N]} \omega_i \log p_i\Big] - \Pi_{T_n} R_p\Big[\log p\Big]. \tag{A.32c}$$

$$= R_p\Big[\sum_{i\in[N]} \omega_i \Pi_{T_n} \log p_i\Big] - R_p\Big[\Pi_{T_n} \log p\Big] \tag{A.32d}$$

Now, multiplying by R_p^{-1} and using the identity $\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(p) = \Pi_{T_n} \log p$ (A.32d) becomes

$$0 = \sum_{i\in[N]} \omega_i \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(p_i) - \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(p). \tag{A.32e}$$

Solving for p establishes the '$\approx$'-relation in (3.17a)

$$p = \exp_{\mathbb{1}_{\mathcal{S}_n}}\left(\sum_{i\in[N]} \omega_i \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(p_i)\right). \tag{A.32f}$$

We obtain the right hand side of (3.17a) by simply plugging the definitions of $\exp_p$ and $\exp_p^{-1}$ into (A.32f). This results in

$$p \overset{(3.17a)}{=} \exp_{\mathbb{1}_{\mathcal{S}_n}}\left(\sum_{i\in[N]} \omega_i \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(p_i)\right) \overset{(3.13b)}{=} \exp_{\mathbb{1}_{\mathcal{S}_n}}\left(\Pi_{T_n} \sum_{i\in[N]} \omega_i \log p_i\right) \tag{A.33a}$$

$$= \exp_{\mathbb{1}_{\mathcal{S}_n}}\left(\Pi_{T_n} \log\Big(\prod_{i\in[N]} p_i^{\omega_i}\Big)\right) \overset{\substack{(3.13a)\\(3.17b)}}{=} \frac{\mathrm{mean}_{g,\omega}(\mathcal{P})}{\langle \mathbb{1}, \mathrm{mean}_{g,\omega}(\mathcal{P})\rangle}, \tag{A.33b}$$

which establishes the formula. □

Proofs of Section 3.2

Proof (Theorem 3.7) The theorem is a direct consequence of Prop. 2.2. First of all, the inverse lifting map $\exp^{-1}_{\mathbb{1}_{\mathcal{W}}} : \mathcal{W} \to \mathcal{T}_{\mathcal{W}}$ at the barycenter $\mathbb{1}_{\mathcal{W}} \in \mathcal{W}$ is a diffeomorphism between the smooth manifolds $\mathcal{W}$ and $\mathcal{T}_{\mathcal{W}}$. By Prop. 2.1 we can construct a unique ϕ-related vector field to X on $\mathcal{T}_{\mathcal{W}}$ with the *pushforward of* X via $\phi = \exp^{-1}_{\mathbb{1}_{\mathcal{W}}}$.

Eq. (3.28b) follows directly as the unique ϕ-related vector field with $\phi = \exp^{-1}_{\mathbb{1}_{\mathcal{W}}}$

$$\dot{V}(t) \overset{(2.6)}{=} d\exp^{-1}_{\mathbb{1}_{\mathcal{W}}}(\exp_{\mathbb{1}_{\mathcal{W}}}(V(t)))\Big[X_{\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))}\Big] \tag{A.34a}$$

$$\overset{(i)}{=} R^{-1}_{\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))}\left[X_{\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))}\right], \tag{A.34b}$$

where we used in (i) the following relation

$$d\exp^{-1}_{\mathbb{1}_{\mathcal{W}}}(W)[X] \overset{(3.14c)}{=} \Pi_T \tfrac{X}{W} \overset{(3.9b)}{=} R^{-1}_W[X]. \tag{A.35}$$

Thus, by construction the vector fields $\dot{W}(t)$ and $\dot{V}(t)$ are ϕ-related. Therefore, a direct application of Prop. 2.2 gives that the integral curves of both vector fields can be transformed into each other via the $\exp_{\mathbb{1}_{\mathcal{W}}}$ map: If $V(t)$ is an integral curve of (3.28b), the curve $\exp_{\mathbb{1}_{\mathcal{W}}} \circ V(t)$ is an integral curve of (3.28a). □

Proof (Corollary 3.8) This corollary is a direct consequence of Theorem. 3.7. By inserting the Riemannian gradient $X_{W(t)} = -\operatorname{grad}_{\mathcal{W}} f(W(t))$ into (3.28b), we obtain (3.30b):

$$\dot{V}(t) \overset{(3.28b)}{=} R^{-1}_{\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))}\left[-\operatorname{grad}_{\mathcal{W}} f(\exp_{\mathbb{1}_{\mathcal{W}}}(V(t)))\right] \tag{A.36a}$$

$$\overset{(3.8)}{=} R^{-1}_{\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))}\left[-R_{\exp_{\mathbb{1}_{\mathcal{W}}} V(t)}\left[\nabla f(\exp_{\mathbb{1}_{\mathcal{W}}}(V(t)))\right]\right] \tag{A.36b}$$

$$= -\nabla f(\exp_{\mathbb{1}_{\mathcal{W}}}(V(t))). \tag{A.36c}$$

The initial point $V(0)$ in (3.30b) follows from

$$V(0) = \exp^{-1}_{\mathbb{1}_{\mathcal{W}}}(W(0)) \overset{(3.13b)}{=} \Pi_T \log(\tfrac{W(0)}{\mathbb{1}_{\mathcal{W}}}) \overset{W(0)=\mathbb{1}_{\mathcal{W}}}{=} \Pi_T \log(1_{m\times n}) = 0_{m\times n}. \tag{A.37}$$

□

Proofs of Section 3.3.2

Proof (Lemma 3.10) With the following relation

$$\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(L_k(W_k)) \overset{(3.41)}{=} \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}\left(\exp_{W_k}(-D_k/\rho)\right) \tag{A.38}$$

$$\overset{(3.14a)}{=} \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}\left(\exp_{\mathbb{1}_{\mathcal{S}_n}}\left(\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(W_k) - D_k/\rho\right)\right) \tag{A.39}$$

$$= \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(W_k) - \Pi_{T_n}(D_k/\rho) \tag{A.40}$$

the similarity matrix (3.42) reads

$$S_i(W) = \exp_{\mathbb{1}_{\mathcal{S}_n}}\Big(\sum_{k\in\mathcal{N}_i} w_{ik} \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(W_k) - \Pi_{T_n}(D_k/\rho)\Big). \tag{A.41}$$

Thus, the corresponding differential factorizes by the chain rule as follows

$$dS_i(W)[V] = d\exp_{\mathbb{1}_{\mathcal{S}_n}}\Big(\sum_{k\in\mathcal{N}_i} w_{ik}\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(L_k(W_k))\Big)\Big[\sum_{k\in\mathcal{N}_i} w_{ik} d\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(W_k)[V_k]\Big] \tag{A.42}$$

$$\overset{(3.14b)}{=} R_{S_i(W)}\Big[\sum_{k\in\mathcal{N}_i} w_{ik} d\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(W_k)[V_k]\Big] \tag{A.43}$$

$$\overset{(3.14c)}{=} R_{S_i(W)}\Big[\sum_{k\in\mathcal{N}_i} w_{ik}\Pi_{T_n}\frac{V_k}{W_k}\Big] \overset{(3.10)}{=} \sum_{k\in\mathcal{N}_i}\omega_{ik} R_{S_i(W)}\left[\frac{V_k}{W_k}\right] \tag{A.44}$$

This establishes formula (3.51). □

A.3 Proofs of Chapter 4

Proofs of Section 4.2

Proof (Proposition 4.2) Let $\gamma\colon(-\varepsilon,\varepsilon)\to\mathcal{W}$ be a smooth curve, with $\varepsilon>0$, $\gamma(0)=\mu_{\mathcal{V}}$ and $\dot\gamma(0)=V$. We then have

$$\langle\nabla E_\tau(\mu_{\mathcal{V}}),V\rangle = \frac{d}{dt}E_\tau\big(\gamma(t)\big)\Big|_{t=0} \tag{A.45a}$$

$$\overset{(4.17)}{=} \sum_{i\in\mathcal{V}}\Big(\langle\Pi_T(\theta_i),V_i\rangle + \sum_{j\colon(i,j)\in\mathcal{E}}\frac{d}{dt}d_{\theta_{ij},\tau}(\gamma_i(t),\gamma_j(t))\Big|_{t=0}\Big), \tag{A.45b}$$

where $\gamma_k(t)$ denotes the k-th row of the matrix $\gamma(t)\in\mathcal{W}\subset\mathbb{R}^{m\times n}$. Since

$$\frac{d}{dt}d_{\theta_{ij},\tau}(\gamma_i(t),\gamma_j(t))\Big|_{t=0} = \langle\nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j),V_i\rangle + \langle\nabla_2 d_{\theta_{ij},\tau}(\mu_i,\mu_j),V_j\rangle\,, \tag{A.46}$$

the r.h.s. of (A.45) becomes

$$\begin{aligned}\langle\nabla E_\tau(\mu_{\mathcal{V}}),V\rangle = \sum_{i\in\mathcal{V}}\Big(\langle\Pi_T(\theta_i),V_i\rangle + \sum_{j\colon(i,j)\in\mathcal{E}}\langle\nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j),V_i\rangle\Big)\\ + \sum_{i\in\mathcal{V}}\sum_{j\colon(i,j)\in\mathcal{E}}\langle\nabla_2 d_{\theta_{ij},\tau}(\mu_i,\mu_j),V_j\rangle\,,\end{aligned} \tag{A.47}$$

where we separated the outer sum over the nodes $i\in\mathcal{V}$ into two parts. Then, the second sum of (A.47) reads

$$\sum_{i\in\mathcal{V}}\sum_{j\colon(i,j)\in\mathcal{E}}\langle\nabla_2 d_{\theta_{ij},\tau}(\mu_i,\mu_j),V_j\rangle = \sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{V}}\delta_{(i,j)\in\mathcal{E}}\langle\nabla_2 d_{\theta_{ij},\tau}(\mu_i,\mu_j),V_j\rangle \tag{A.48a}$$

$$= \sum_{j\in\mathcal{V}} \sum_{i\in\mathcal{V}} \delta_{(i,j)\in\mathcal{E}} \langle \nabla_2 d_{\theta_{ij},\tau}(\mu_i,\mu_j), V_j\rangle \tag{A.48b}$$

$$= \sum_{j\in\mathcal{V}} \sum_{i\colon (i,j)\in\mathcal{E}} \langle \nabla_2 d_{\theta_{ij},\tau}(\mu_i,\mu_j), V_j\rangle \tag{A.48c}$$

$$= \sum_{i\in\mathcal{V}} \sum_{j\colon (j,i)\in\mathcal{E}} \langle \nabla_2 d_{\theta_{ji},\tau}(\mu_j,\mu_i), V_i\rangle\,, \tag{A.48d}$$

where $\delta_{(k,l)\in\mathcal{E}}$ is the indicator function with value 1 if $(k,l)\in\mathcal{E}$ and 0 if $(k,l)\notin\mathcal{E}$. The last equation (A.48d) follows by renaming the indices of summation. Substitution into (A.47) gives

$$\langle \nabla E_\tau(\mu_\mathcal{V}), V\rangle = \sum_{i\in\mathcal{V}} \Big\langle \Pi_T(\theta_i) + \sum_{j\colon (i,j)\in\mathcal{E}} \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j) + \sum_{j\colon (j,i)\in\mathcal{E}} \nabla_2 d_{\theta_{ji},\tau}(\mu_j,\mu_i), V_i\Big\rangle \tag{A.49a}$$

$$= \sum_{i\in\mathcal{V}} \langle \nabla_i E_\tau(\mu_\mathcal{V}), V_i\rangle \tag{A.49b}$$

which proves (4.18). □

Proof (Lemma 4.3) Let $M_*\in\Pi(\mu_i,\mu_j)$ be a minimizer of (4.12). Then, due to the assumption on F_τ, we have

$$d_{\theta_{ij},\tau}(\mu_i,\mu_j) = \langle \theta_{ij}, M_*\rangle + F_\tau(M_*) = \langle \theta_{ij}^\top, M_*^\top\rangle + F_\tau(M_*^\top)\,. \tag{A.50}$$

Now, let $\tilde{M}\in\Pi(\mu_j,\mu_i)$ be an arbitrary coupling measure. Then, $\tilde{M}^\top\in\Pi(\mu_i,\mu_j)$ and we have

$$\langle \theta_{ij}^\top, \tilde{M}\rangle + F_\tau(\tilde{M}) = \langle \theta_{ij}, \tilde{M}^\top\rangle + F_\tau(\tilde{M}^\top) \geq \langle \theta_{ij}, M_*\rangle + F_\tau(M_*) = \langle \theta_{ij}^\top, M_*^\top\rangle + F_\tau(M_*^\top)\,. \tag{A.51}$$

This shows that $M_*^\top\in\Pi(\mu_j,\mu_i)$ is a minimizer of $d_{\theta_{ij}^\top,\tau}(\mu_j,\mu_i)$ and establishes equation (4.20). □

Proof (Corollary 4.4) Due to Lemma 4.3, we have $\nabla_2 d_{\theta_{ji},\tau}(\mu_j,\mu_i) = \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j)$ and obtain

$$\nabla_i E_\tau(\mu_\mathcal{V}) \overset{(4.18)}{=} \Pi_T(\theta_i) + \sum_{j\colon (i,j)\in\mathcal{E}} \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j) + \sum_{j\colon (j,i)\in\mathcal{E}} \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j) \tag{A.52a}$$

$$= \Pi_T(\theta_i) + \sum_{j\in\mathcal{N}(i)} \nabla_1 d_{\theta_{ij},\tau}(\mu_i,\mu_j), \tag{A.52b}$$

which proves (4.22). □

Proofs of Section 4.3

Proof (Theorem 4.5) The proof is divided into three steps:

1. Relate the orthogonal decomposition $\mathbb{R}^{2n} = \ker(\mathcal{A}^\top) \oplus \operatorname{im}(\mathcal{A})$ to the tangent space $T_p(\mathcal{S}\times\mathcal{S}) = T \times T \subset \mathbb{R}^{2n}$ for any $p = (p_1, p_2) \in \mathcal{S}\times\mathcal{S}$.
2. Show the existence of a global isometric chart for the manifold $\mathcal{S}\times\mathcal{S}$. The goal is to represent the smoothed Wasserstein distance $d_{\Theta,\tau}$ and the dual objective function $g(\mu,\nu)$ in a convenient way.
3. End the proof by applying Theorem 1.2.

We proceed by subsequently doing the steps mentioned above:

1. Consider the unique decomposition $v = v_{\ker} + v_{\operatorname{im}} \in \ker(\mathcal{A}^\top) \oplus \operatorname{im}(\mathcal{A})$ of any point $v \in \mathbb{R}^{2n}$. Then we have

$$\Pi_{T\times T}(v_{\operatorname{im}}) = v_T = \Pi_{T\times T}(v). \tag{A.53}$$

   At first, we show $T \times T \subseteq \operatorname{im}(\mathcal{A})$. For this, take an arbitrary $v = \binom{v_i}{v_j} \in T \times T$. Due to the definition of T, we have $\langle \mathbb{1}_n, v_i\rangle = \langle \mathbb{1}_n, v_j\rangle = 0$ and thus $\langle v, \binom{\mathbb{1}_n}{-\mathbb{1}_n}\rangle = 0$, which according to Lemma 4.7 means $v \in \ker(\mathcal{A}^\top)^\perp = \operatorname{im}(\mathcal{A})$. As a consequence of $T \times T \subseteq \operatorname{im}(\mathcal{A})$, we have $\Pi_{T\times T}(v_{\ker}) = 0$ and therefore (A.53) follows from

$$\Pi_{T\times T}(v) - \Pi_{T\times T}(v_{\operatorname{im}}) = \Pi_{T\times T}(v - v_{\operatorname{im}}) = \Pi_{T\times T}(v_{\ker}) = 0. \tag{A.54}$$

2. There exists an open subset $U \subset \mathbb{R}^{2(n-1)}$ and an isometry $\phi\colon U \to \mathcal{S}\times\mathcal{S}$ such that ϕ^{-1} is a global isometric chart of the manifold $\mathcal{S}\times\mathcal{S}$. ϕ can be constructed as follows. Choose an orthonormal basis $\{v_i, \dots, v_{2(n-1)}\}$ of the tangent space $T \times T$, set $b = \frac{1}{n}\binom{\mathbb{1}_n}{\mathbb{1}_n}$ and define the isometry

$$\psi\colon \mathbb{R}^{2(n-1)} \to \big(T \times T\big) + b, \quad x \mapsto \psi(x) := Bx + b, \quad Bx = \sum_{i=1}^{2(n-1)} x_i v_i. \tag{A.55}$$

   Because $\mathcal{S}\times\mathcal{S}$ is an open subset of $\big(T \times T\big) + b$ and ψ an isometry, we have that the set $U := \psi^{-1}(\mathcal{S}\times\mathcal{S}) \subset \mathbb{R}^{2(n-1)}$ is also open and

$$\phi := \psi|_U\colon U \to \mathcal{S}\times\mathcal{S} \tag{A.56}$$

the desired isometric mapping. Furthermore, since the basis $\{v_i\}_{i=1}^{2(n-1)}$ is orthonormal, the orthogonal projection reads

$$\Pi_{T\times T} = BB^\top. \tag{A.57}$$

3. Using ϕ given by (A.56), we obtain the coordinate representations

$$\overline{d}_{\Theta,\tau} := d_{\Theta,\tau} \circ \phi, \qquad \overline{g}(x,v) := g\big(\phi(x), v\big) \tag{A.58}$$

of the smoothed Wasserstein distance $d_{\Theta,\tau}$ and the dual objective function $g(p,v)$. Since we assume strong duality, that is equality of the optimal values of (4.26) and (4.27), we have $d_{\Theta,\tau}(p) = \max_{v\in\mathbb{R}^{2n}} g(p,v)$. Setting $x_p = \phi^{-1}(p)$, this equation translates in view of Lemma 4.10 to

$$\overline{g}(x_p, \overline{v}_{\mathrm{im}}) = \max_{v_{\mathrm{im}}\in \mathrm{im}(\mathcal{A})} \overline{g}(x_p, v_{\mathrm{im}}) = \overline{g}(x_p, \overline{v}) = \max_{v\in\mathbb{R}^{2n}} \overline{g}(x_p, v) = \overline{d}_{\Theta,\tau}(x_p), \tag{A.59}$$

with unique maximizer $\overline{v}_{\mathrm{im}} = \Pi_{\mathrm{im}(\mathcal{A})}(\overline{v})$. Let $\mathbb{B}_\delta \subset \mathrm{im}(\mathcal{A})$ be a compact neighborhood of $\overline{v}_{\mathrm{im}}$. Then (A.59) remains valid after restricting $\mathrm{im}(\mathcal{A})$ to $\mathbb{B}_\delta$. Because g, given by (4.23), is linear in the first argument and the mapping ϕ is affine, the function $\overline{g}$ is convex in the first argument and differentiable. Hence $\overline{g}$ satisfies the assumptions of Theorem 1.2.

In order to compute the gradient $\nabla_x \overline{g}(x, v_{\mathrm{im}})$, it suffices to consider the first term $\langle \phi(x), v_{\mathrm{im}}\rangle$ of $\overline{g}$, which only depends on x. Using (A.56), we have

$$\langle \phi(x), v_{\mathrm{im}}\rangle = \langle Bx + b, v_{\mathrm{im}}\rangle = \langle x, B^\top v_{\mathrm{im}}\rangle + \langle b, v_{\mathrm{im}}\rangle. \tag{A.60}$$

Thus, $\nabla_x \overline{g}(x, v_{\mathrm{im}}) = B^\top v_{\mathrm{im}}$ which continuously depends on v_{im}. As a consequence, we may apply Theorem 1.2 and obtain due to (1.17)

$$\nabla \overline{d}_{\Theta,\tau}(x_p) = \nabla_x \overline{g}(x_p, \overline{v}_{\mathrm{im}}) = B^\top \overline{v}_{\mathrm{im}}. \tag{A.61}$$

Using the differential $D\phi(x) = B$, we finally get

$$\nabla d_{\Theta,\tau}(p) = B\nabla \overline{d}_{\Theta,\tau}(x_p) = BB^\top \overline{v}_{\mathrm{im}} \overset{(A.57)}{=} \Pi_{T\times T}(\overline{v}_{\mathrm{im}}) \overset{(A.53)}{=} \overline{v}_T, \tag{A.62}$$

which proves (4.24). □

Proof (Lemma 4.6) Taking into account (2.41b), we write the right-hand side of (4.12) in the form

$$\min_{M\in\mathbb{R}^{n\times n}} \langle \Theta, M\rangle + G_\tau(M) \quad \text{s.t.} \quad \mathcal{A}M = \mu, \quad M \geq 0. \tag{A.63}$$

Let $\nu = (\nu_1, \nu_2) \in \mathbb{R}^{2n}$ denote the dual variables corresponding to the affine constraint of (A.63). Then problem (A.63) rewritten in Lagrangian form reads

$$\min_{M\in\mathbb{R}^{n\times n}} \left\{\langle \Theta, M\rangle + G_\tau(M) + \max_\nu \langle \nu, \mu - \mathcal{A}M\rangle\right\} \tag{A.64a}$$

$$\Leftrightarrow \quad \min_{M\in\mathbb{R}^{n\times n}} \left\{\max_\nu \langle \nu, \mu\rangle + G_\tau(M) - \langle \mathcal{A}^\top \nu - \Theta, M\rangle\right\}. \tag{A.64b}$$

Since strong duality holds by assumption, interchanging min and max yields the dual problem (4.27). Moreover, the optimal primal and dual objective function values are equal, which gives with (A.64a) and (4.27)

$$-\langle \overline{M}, \mathcal{A}^\top \overline{\nu} - \Theta\rangle + G_\tau(\overline{M}) + G_\tau^*(\mathcal{A}^\top \overline{\nu} - \Theta) = 0. \tag{A.65}$$

This implies (4.28a) by the subgradient inversion rule (see Theorem 1.1), whereas the primal constraint (4.28b) is obvious. □

Proof (Lemma 4.7) Let $z = \binom{x}{y} \in \mathbb{R}^{2n}$ with $z \in \ker(\mathcal{A}^\top)$. Applying $\mathcal{A}$, we get

$$0 = \mathcal{A}\mathcal{A}^\top z \overset{(1.11b)}{=} \mathcal{A}(x\mathbb{1}^\top) + \mathcal{A}(\mathbb{1}y^\top) \overset{(1.11a)}{=} \begin{pmatrix} nx + \langle y, \mathbb{1}_n\rangle \mathbb{1}_n \\ \langle x, \mathbb{1}_n\rangle \mathbb{1}_n + ny \end{pmatrix} \tag{A.66}$$

and solving this equation to z gives (4.29a)

$$z = \begin{pmatrix} x \\ y \end{pmatrix} = -\frac{1}{n}\begin{pmatrix} \langle y, \mathbb{1}_n\rangle \mathbb{1}_n \\ \langle x, \mathbb{1}_n\rangle \mathbb{1}_n \end{pmatrix} \overset{(i)}{=} \lambda \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix}, \quad \text{with } \lambda := \frac{1}{n}\langle x, \mathbb{1}_n\rangle \in \mathbb{R}. \tag{A.67}$$

Note, that we have exploited in step (i) that $\langle x, \mathbb{1}_n\rangle = -\langle y, \mathbb{1}_n\rangle$ applies. Conversely, in view of the definition (1.11b), it is clear that any vector from the set (4.29a) is in $\ker(\mathcal{A}^\top)$. The characterization of $\ker(\mathcal{A}^\top)^\perp$ directly follows from the definitions. □

Proof (Lemma 4.8) We first show „⊆“ in (4.30b), i.e. if $\overline{\nu}$ is an optimal dual solution, then

$$\operatorname*{argmax}_{\nu\in\mathbb{R}^{2n}} g(p, \nu) \subseteq \overline{\nu} + \ker(\mathcal{A}^\top). \tag{A.68}$$

Let $\overline{v}' \neq \overline{v}$ be another optimal dual solution, that is $g(p,\overline{v}) = g(p,\overline{v}')$. By (4.23), this equation reads

$$G_\tau^*(\mathcal{A}^\top \overline{v} - \Theta) - G_\tau^*(\mathcal{A}^\top \overline{v}' - \Theta) = \langle p, \overline{v} - \overline{v}' \rangle \,. \tag{A.69}$$

Moreover, due to the optimality conditions (4.28), $\overline{v}'$ satisfies

$$\overline{M}' = \nabla G_\tau^*(\mathcal{A}^\top \overline{v}' - \Theta), \qquad \mathcal{A}\overline{M}' = p, \tag{A.70}$$

where $\overline{M}'$ is the corresponding primal optimal solution. Hence

$$\langle p, \overline{v} - \overline{v}' \rangle = \langle \mathcal{A}\overline{M}', \overline{v} - \overline{v}' \rangle = \langle \overline{M}', \mathcal{A}^\top(\overline{v} - \overline{v}') \rangle \overset{(A.70)}{=} \langle \nabla G_\tau^*(\mathcal{A}^\top \overline{v}' - \Theta), \mathcal{A}^\top(\overline{v} - \overline{v}') \rangle \,. \tag{A.71}$$

Using the shorthands $\overline{w} = \mathcal{A}^\top \overline{v} - \Theta$ and $\overline{w}' = \mathcal{A}^\top \overline{v}' - \Theta$, we have

$$\overline{w}' - \overline{w} = \mathcal{A}^\top(\overline{v}' - \overline{v}) \tag{A.72}$$

and therefore

$$G_\tau^*(\overline{w}') - G_\tau^*(\overline{w}) \overset{(A.69)}{=} \langle p, \overline{v}' - \overline{v} \rangle \overset{(A.71)}{=} \langle \nabla G_\tau^*(\overline{w}'), \overline{w}' - \overline{w} \rangle. \tag{A.73}$$

Since G_τ^* is strictly convex, this equality can only hold if

$$0 = \overline{w}' - \overline{w} \overset{(A.72)}{=} \mathcal{A}^\top(\overline{v}' - \overline{v}) \implies \overline{v}' - \overline{v} \in \ker(\mathcal{A}^\top). \tag{A.74}$$

This shows that $\overline{v}$ and $\overline{v}'$ can only differ by a nullspace vector, i.e. we have shown relation (A.68).

Next, we show „$\supseteq$“ in (4.30b), that is vectors characterized by the right-hand side of (4.30b) maximize the dual objective function $g(p,v)$. Let again $\overline{v}$ be an optimal dual solution, and let $\overline{v}' \in \overline{v} + \ker(\mathcal{A}^\top)$ be an arbitrary vector. Lemma 4.7 implies that $\overline{v}'$ takes the form

$$\overline{v}' = \overline{v} + \lambda \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix}, \qquad \lambda \in \mathbb{R}. \tag{A.75}$$

Then, since $\left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle = 0$ and $\mathcal{A}^\top \overline{v}' = \mathcal{A}^\top \overline{v}$, we have

$$g(a,\overline{v}') = \langle p, \overline{v} + \lambda \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \rangle - G_\tau^*\left(\mathcal{A}^\top\left(\overline{v} + \lambda \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix}\right) - \Theta\right) \tag{A.76a}$$

$$= \langle p, v \rangle - G_\tau^*(\mathcal{A}^\top \overline{v} - \Theta) = g(a,\overline{v}), \tag{A.76b}$$

that is $\overline{v}' \in \operatorname{argmax}_{v\in\mathbb{R}^{2n}} g(p,v)$.

Finally, we show (4.30a): Suppose $\left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle \neq 0$, $\overline{v}$ is an optimal dual solution and $\overline{v}'$ is another optimal dual vector of the form (A.75) shown above. Inserting (A.75) into (A.69) yields

$$0 = \langle p, \overline{v}' - \overline{v} \rangle = \lambda \langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \rangle \implies \lambda = 0, \tag{A.77}$$

since $\left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle \neq 0$. Thus, $\overline{v}' = \overline{v}$ by (A.75), which shows uniqueness of $\overline{v}$ as claimed by (4.30a). □

Proof (Lemma 4.9) We first recall that for any subspace $U \subset V$ of a finite dimensional vector space V the orthogonal decomposition $V = U \oplus U^{\perp}$ applies.

Now, let $F\colon V \to W$ be linear map between finite dimensional vector spaces. The statement follows by the basic linear algebra formula

$$\operatorname{im}(F) = \ker(F^{\top})^{\perp}. \tag{A.78}$$

By setting $V = \mathbb{R}^{2n}$ and $U = \ker(\mathcal{A}^{\top})$ we have

$$\mathbb{R}^{2n} = \ker(\mathcal{A}^{\top}) \oplus \ker(\mathcal{A}^{\top})^{\perp} \overset{(A.78)}{=} \ker(\mathcal{A}^{\top}) \oplus \operatorname{im}(\mathcal{A}), \tag{A.79}$$

which proofs statement (4.31). □

Proof (Lemma 4.10) We first show (4.32b). Let $\overline{v}$ be an optimal dual solution. Since $\left\langle p, \begin{pmatrix} \mathbb{1}_n \\ -\mathbb{1}_n \end{pmatrix} \right\rangle = 0$, (4.30b) of Lemma 4.8 yields

$$\operatorname{argmax}_{v\in\mathbb{R}^{2n}} g(p,v) = \overline{v} + \ker(\mathcal{A}^{\top}) = \overline{v}_{\ker} + \overline{v}_{\operatorname{im}} + \ker(\mathcal{A}^{\top}).$$

This shows $\overline{v}_{\operatorname{im}} \in \overline{v} + \ker(\mathcal{A}^{\top})$, that is $\overline{v}_{\operatorname{im}} \in \operatorname{im}(\mathcal{A})$ is a maximizer, which implies (4.32b).

Now, we show (4.32a). Let $\overline{v}'_{\operatorname{im}} \in \operatorname{im}(\mathcal{A})$ be another maximizer. As before, we use the representation $\overline{v}'_{\operatorname{im}} \in \overline{v} + \ker(\mathcal{A}^{\top})$, that is $\overline{v}'_{\operatorname{im}} = \overline{v}_{\ker} + \overline{v}_{\operatorname{im}} + \tilde{v}_{\ker}$, for some $\tilde{v}_{\ker} \in \ker(\mathcal{A}^{\top})$. This implies $\overline{v}'_{\operatorname{im}} = \overline{v}_{\operatorname{im}}$, i.e. uniqueness (4.32a) of the dual maximizer in $\operatorname{im}(\mathcal{A})$. □

Proofs of Section 4.4.1

Proof (Proposition 4.12) An Euler-step for following the Riemannian gradient descent flow of $f_{\tau,\alpha}$ on the tangent space reads (with $\nabla_i = \nabla_{W_i}$)

$$V_i^{(k+1)} = V_i^{(k)} - h\nabla_i f(W^{(k)}) = V_i^{(k)} - h\nabla_i E_\tau(W^{(k)}) - \alpha\nabla_i H(W^{(k)}), \qquad i \in [m], \tag{A.80}$$

where the i-th row of $W^{(k)}$ is given by $W_i^{(k)} = \exp_{\mathbb{1}_{\mathcal{S}_n}}(V_i^{(k)})$, $\mathbb{1}_{\mathcal{S}_n} = \frac{1}{n}\mathbb{1}_n$.

In order to compute the gradient of the entropy, we consider a smooth curve $\gamma\colon (-\varepsilon,\varepsilon) \to \mathcal{W}$ with $\gamma(0) = W$ and $\dot\gamma(0) = X$. Then

$$\frac{d}{dt} H(\gamma(t))|_{t=0} = -\langle X, \log(W)\rangle - \langle W, \frac{1}{W}\cdot X\rangle = -\langle X, \log(W)\rangle - \langle \mathbb{1}\mathbb{1}^\top, X\rangle. \tag{A.81}$$

Since $\langle \log(W), X\rangle = \langle \Pi_{T^m}\left(\log(W)\right), X\rangle$ and $\langle \mathbb{1}\mathbb{1}^\top, X\rangle = \langle \mathbb{1}, X\mathbb{1}\rangle = \langle \mathbb{1}, 0\rangle = 0$, we have

$$\langle \nabla H(W), X\rangle = \frac{d}{dt} H(\gamma(t))|_{t=0} = \langle -\Pi_{T^m}\left(\log(W)\right), X\rangle. \tag{A.82}$$

Thus, using $\Pi_T(\log(W_i)) = \exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}(W_i)$ from (3.13b), we obtain

$$\nabla_i H(W^{(k)}) = -\Pi_T\left(\log(W_i^{(k)})\right) = -\exp^{-1}_{\mathbb{1}_{\mathcal{S}_n}}\left(\exp_{\mathbb{1}_{\mathcal{S}_n}}(V_i^{(k)})\right) = -V_i^{(k)}. \tag{A.83}$$

Substitution into (A.80) gives

$$V_i^{(k+1)} = (1+\alpha)V_i^{(k)} - h\nabla_i E_\tau(W^{(k)}) \tag{A.84}$$

and in turn the update

$$W_i^{(k+1)} = \exp_{\mathbb{1}_{\mathcal{S}_n}}(V_i^{(k+1)}) = \frac{e^{(1+\alpha)V_i^{(k)}}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle \mathbb{1}_n, e^{(1+\alpha)V_i^{(k)}}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}\rangle} \tag{A.85a}$$

$$= \frac{(e^{V_i^{(k)}})^{(1+\alpha)}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle \mathbb{1}_n, (e^{V_i^{(k)}})^{1+\alpha}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}\rangle} = \frac{(W_i^{(k)})^{(1+\alpha)}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle \mathbb{1}_n, (W_i^{(k)})^{1+\alpha}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}\rangle} \tag{A.85b}$$

$$= \frac{(W_i^{(k)})^{(1+\alpha)}\cdot e^{-h\nabla_i E_\tau(W^{(k)})}}{\langle (W_i^{(k)})^{1+\alpha}, e^{-h\nabla_i E_\tau(W^{(k)})}\rangle} \tag{A.85c}$$

which is (4.44). □

A.4 Proofs of Chapter 5

Proofs of Section 5.2

Proof (Proposition 5.1) Let $V \in \mathcal{T}_{\mathcal{W}}$. Note that for every $i \in \mathcal{V}$

$$\langle W_i^*, \log\big(\exp_{\mathbb{1}_{S_n} n}(V_i)\big)\rangle = \langle W_i^*, V_i - \log(\langle \mathbb{1}, e^{V_i}\rangle)\mathbb{1}\rangle = \langle W_i^*, V_i\rangle + \log(\langle \mathbb{1}, e^{V_i}\rangle). \quad \text{(A.86)}$$

Hence the KL-divergence between W_i^* and the induced assignment $W_i = \exp_{\mathbb{1}_{S_n} n}(V_i)$ takes the form

$$\mathrm{KL}\big(W_i^*, W_i\big) = \langle W_i^*, \log(W_i^*)\rangle - \langle W_i^*, V_i\rangle + \log(\langle \mathbb{1}, e^{V_i}\rangle) \quad \text{(A.87)}$$

and results in the following expression for $\mathcal{C}$ from (5.8),

$$\mathcal{C}(V) = \langle W^*, \log(W^*)\rangle - \langle W^*, V\rangle + \sum_{i\in[m]} \log(\langle \mathbb{1}, e^{V_i}\rangle). \quad \text{(A.88)}$$

Take $X \in \mathbb{R}^{m\times n}$ and set $\gamma(t) := V + tX$ for $t \in \mathbb{R}$. Then, the above formula for $\mathcal{C}$ implies

$$\langle \partial C(V), X\rangle = \frac{d}{dt} C(\gamma(t))|_{t=0} = -\langle W^*, X\rangle + \sum_{i\in[m]} \frac{1}{\langle \mathbb{1}, e^{V_i}\rangle} \langle e^{V_i}, X_i\rangle = \langle \exp_{\mathbb{1}_{\mathcal{W}}}(V) - W^*, X\rangle. \quad \text{(A.89)}$$

Since $X \in \mathbb{R}^{m\times n}$ was arbitrary, the expression (5.10) follows. □

Proof (Lemma 5.2) For arbitrary $B \in \mathbb{R}^{m\times m}$ and $\Omega \in \mathbb{R}^{m\times N}$, we obtain $\langle A_\Omega, B\rangle = \sum_{i,j\in\mathcal{V}} \delta_{j\in\mathcal{N}_i} \Omega_{ik} B_{ik} = \langle \Omega, A_B^\top\rangle$ due to (5.11). □

Proof (Proposition 5.3) Let $V, X \in \mathcal{T}_{\mathcal{W}}$ and set $\gamma(t) := V + tX \in \mathcal{T}_0$ for all $t \in \mathbb{R}$. Then

$$d_V F(V,\Omega)[X] = \frac{d}{dt} F(\gamma(t),\Omega)|_{t=0} = R_{S(W_0)}[A_\Omega \dot\gamma(0)] = R_{S(W_0)}[A_\Omega X]. \quad \text{(A.90)}$$

Similarly, for $\Omega \in \mathcal{P}$ and $\Psi \in \mathcal{T}_{\mathcal{P}}$, let $\eta(t) := \Omega + t\Psi \in \mathcal{P}$ be a curve with $t \in (-\varepsilon, \varepsilon)$ for sufficiently small $\varepsilon > 0$. The linearity of the averaging operator A_Ω with respect to Ω gives

$$d_\Omega F(V,\Omega)[X] = \frac{d}{dt} F(V,\eta(t))|_{t=0} = \frac{d}{dt} R_{S(W_0)}[A_{\eta(t)} V]|_{t=0} = R_{S(W_0)} A_\Psi[V]. \quad \text{(A.91)}$$

We now determine the adjoint differentials. Consider arbitrary $X, Y \in \mathcal{T}_{\mathcal{W}}$ and note that the linear map $R_S(W_0)$ is symmetric, since every component map $R_{S_i(W_0)}$ is symmetric by (3.7). Thus,

$$\langle d_V F(V,\Omega)[Y], X\rangle = \langle R_{S(W_0)}[A_\Omega Y], X\rangle = \langle Y, A_\Omega^\top R_{S(W_0)}[X]\rangle \tag{A.92}$$

and therefore $d_V F(V,\Omega)^\top[X] = A_\Omega^\top R_{S(W_0)}[X]$. Now let arbitrary $\Psi \in \mathcal{T}_{\mathcal{P}}$ and $X \in \mathcal{T}_{\mathcal{P}}$ be given. Then

$$\begin{aligned}\langle d_\Omega F(V,\Omega)[\Psi], X\rangle &= \langle R_{S(W_0)}[A_\Psi V], X\rangle = \langle A_\Psi, (R_{S(W_0)}[X])V^\top\rangle &\text{(A.93a)}\\ &= \langle \Psi, A^\top_{(R_{S(W_0)}[X])V^\top}\rangle = \langle \Psi, \Pi_{\mathcal{P}}\big[A^\top_{(R_{S(W_0)}[X])V^\top}\big]\rangle, &\text{(A.93b)}\end{aligned}$$

which proves the expression for the corresponding adjoint. □

Appendix B

Derivation of Loopy Belief Propagation

As stated in Section 2.2.5, *loopy belief propagation* by *message passing* is given by the fixed point equation (2.50). In this section we derive equation (2.50) in detail.

We start by considering the *smoothed* primal linear program (2.46) written in the form

$$\min_{\mu} \quad \langle \theta, \mu \rangle - \varepsilon H(\mu) \tag{B.1a}$$

$$\text{s.t.} \quad \sum_{x_i \in \mathcal{X}} \mu_i(x_i) = 1, \qquad \forall i \in \mathcal{V}, \tag{B.1b}$$

$$\sum_{x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) = \mu_i, \qquad \forall ij \in \mathcal{E}, \forall x_i \in \mathcal{X}, \tag{B.1c}$$

$$\sum_{x_i \in \mathcal{X}} \mu_{ij}(x_i, x_j) = \mu_j, \qquad \forall ij \in \mathcal{E}, \forall x_j \in \mathcal{X}, \tag{B.1d}$$

$$\mu \geq 0, \tag{B.1e}$$

with smoothing parameter $\varepsilon > 0$, *Bethe entropy* function

$$H(\mu) = \sum_{ij \in \mathcal{E}} H(\mu_{ij}) - \sum_{i \in \mathcal{V}} \big(d(i) - 1\big) H(\mu_i), \tag{B.2a}$$

degree $d(i) := |\mathcal{N}(i)|$ of vertex i and local entropy functions

$$H(\mu_i) = -\sum_{x_i \in \mathcal{X}} \mu_i(x_i) \log \mu_i(x_i), \tag{B.2b}$$

$$H(\mu_{ij}) = -\sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \mu_{ij}(x_i, x_j). \tag{B.2c}$$

The constraints (B.1b)-(B.1d) represent the feasible set $\mathcal{L}_{\mathcal{G}}$.

As already pointed out in Section 2.2.5, we obtain a relation between μ and ν by evaluating the optimality condition $\nabla_{\mu} L(\mu, \nu) = 0$ based on the corresponding Lagrangian of (B.1). In order to derive explicit formulas, we rewrite the *Bethe entropy* function (B.2a) in terms of the *mutual information* which is summarized in the following lemma.

Lemma B.1
Suppose μ satisfies the marginalization constraints (2.39). Then, the *Bethe entropy* (B.2a) can be rewritten in the form

$$H(\mu) = \sum_{i \in \mathcal{V}} H(\mu_i) - \sum_{ij \in \mathcal{E}} I_{ij}(\mu_{ij}) \tag{B.3}$$

where I_{ij} is the *mutual information*

$$I_{ij}(\mu_{ij}) := \mathrm{KL}\big(\mu_{ij}, \mu_i \mu_j\big) = \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)}. \tag{B.4}$$

Proof The proof is straightforward. First, we rewrite the mutual information

$$\begin{aligned}
I_{ij}(\mu_{ij}) &\overset{(B.4)}{=} \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)} && \text{(B.5)} \\
&= \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \big(\log \mu_{ij}(x_i, x_j) - \log \mu_i(x_i) - \log \mu_j(x_j)\big) && \text{(B.6)} \\
&= \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \mu_{ij}(x_i, x_j) \\
&\quad - \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \mu_i(x_i) - \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \mu_j(x_j). && \text{(B.7)}
\end{aligned}$$

The last two summands can be rewritten as follows

$$\begin{aligned}
- \sum_{x_i, x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) \log \mu_i(x_i) &= - \sum_{x_i \in \mathcal{X}} \log \mu_i(x_i) \sum_{x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) && \text{(B.8)} \\
&\overset{(i)}{=} - \sum_{x_i \in \mathcal{X}} \log \mu_i(x_i) \mu_i(x_i) \overset{(B.2b)}{=} H(\mu_i). && \text{(B.9)}
\end{aligned}$$

In (i) we have taken into account that the marginalization constraints (2.39) have to hold. Consequently, (B.7) reads

$$I_{ij}(\mu_{ij}) \overset{(B.2c)}{=} -H(\mu_{ij}) + H(\mu_i) + H(\mu_j). \tag{B.10}$$

Inserting (B.10) into (B.3) gives

$$\begin{aligned}
H(\mu) &= \sum_{i \in \mathcal{V}} H(\mu_i) - \sum_{ij \in \mathcal{E}} I_{ij}(\mu_{ij}) && \text{(B.11a)} \\
&= \sum_{i \in \mathcal{V}} H(\mu_i) + \sum_{ij \in \mathcal{E}} H(\mu_{ij}) - \sum_{ij \in \mathcal{E}} H(\mu_i) - \sum_{ij \in \mathcal{E}} H(\mu_j) && \text{(B.11b)}
\end{aligned}$$

and by plugging the following identity

$$\sum_{ij\in\mathcal{E}} H(\mu_j) = \tfrac{1}{2}\sum_{i\in\mathcal{V}}\sum_{j\in\mathcal{N}(i)} H(\mu_i) = \tfrac{1}{2}\sum_{i\in\mathcal{V}} H(\mu_i)\sum_{j\in\mathcal{N}(i)} 1 = \tfrac{1}{2}\sum_{i\in\mathcal{V}} H(\mu_i)d(i) \tag{B.11c}$$

into (B.11b) gives

$$\begin{aligned} H(\mu) &= \sum_{i\in\mathcal{V}} H\big(\mu_i\big) + \sum_{ij\in\mathcal{E}} H(\mu_{ij}) - \tfrac{1}{2}\sum_{i\in\mathcal{V}} H(\mu_i)d(i) - \tfrac{1}{2}\sum_{j\in\mathcal{V}} H(\mu_j)d(j) \\ &= \sum_{i\in\mathcal{V}} H\big(\mu_i\big) + \sum_{ij\in\mathcal{E}} H(\mu_{ij}) - \sum_{i\in\mathcal{V}} H(\mu_i)d(i) \qquad \text{(B.11d)} \\ &= \sum_{ij\in\mathcal{E}} H(\mu_{ij}) - \sum_{i\in\mathcal{V}} \big(d(i)-1\big)H(\mu_i). \qquad \text{(B.11e)} \end{aligned}$$

This establishes formula (B.2a). □

Now, in order to find an expression for the derivative of the *Bethe entropy* we compute

$$\begin{aligned} \frac{d}{d\mu_i(x_i)} I_{ij}(\mu_{ij}) &\overset{(B.4)}{=} \frac{d}{d\mu_i(x_i)}\left(\sum_{x_i,x_j\in\mathcal{X}} \mu_{ij}(x_i,x_j)\log\frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)}\right) && \text{(B.12a)} \\ &= \sum_{x_j\in\mathcal{X}} \mu_i(x_i)\mu_j(x_j)\frac{d}{d\mu_i(x_i)}\left(\frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)}\right) && \text{(B.12b)} \\ &= \sum_{x_j\in\mathcal{X}} \mu_i(x_i)\mu_{ij}(x_i,x_j)\left(-\frac{1}{\mu_i^2(x_i)}\right) \overset{(ii)}{=} -1, && \text{(B.12c)} \end{aligned}$$

whereby in (ii) we again have taken into account that the marginalization constraints (2.39) have to hold, and

$$\frac{d}{d\mu_{ij}(x_i,x_j)} I_{ij}(\mu_{ij}) \overset{(B.4)}{=} \frac{d}{d\mu_{ij}(x_i,x_j)}\left(\sum_{x_i,x_j\in\mathcal{X}} \mu_{ij}(x_i,x_j)\log\frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)}\right) \tag{B.12d}$$

$$\begin{aligned} &= \log\frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)} + \mu_i(x_i)\mu_j(x_j)\frac{1}{\mu_i(x_i)\mu_j(x_j)} && \text{(B.12e)} \\ &= \log\frac{\mu_{ij}(x_i,x_j)}{\mu_i(x_i)\mu_j(x_j)} + 1. && \text{(B.12f)} \end{aligned}$$

By using

$$\frac{d}{d\mu_i(x_i)} H(\mu_i) \overset{(B.2b)}{=} \frac{d}{d\mu_i(x_i)} \left(- \sum_{x_i \in \mathcal{X}} \mu_i(x_i) \log \mu_i(x_i) \right) = -\log \mu_i(x_i) - 1, \tag{B.13}$$

the derivatives of the *Bethe entropy* are given by

$$\frac{d}{d\mu_i(x_i)} H(\mu) \overset{(B.3)}{=} \frac{d}{d\mu_i(x_i)} \left(\sum_{i \in \mathcal{V}} H(\mu_i) \right) - \frac{d}{d\mu_i(x_i)} \left(\sum_{ij \in \mathcal{E}} I_{ij}(\mu_{ij}) \right) \tag{B.14a}$$

$$\overset{\substack{(B.12c) \\ (B.13)}}{=} -\log \mu_i(x_i) - 1 + 1 = -\log \mu_i(x_i) \tag{B.14b}$$

$$\frac{d}{d\mu_{ij}(x_i, x_j)} H(\mu) \overset{(B.3)}{=} - \frac{d}{d\mu_{ij}(x_i, x_j)} \left(\sum_{ij \in \mathcal{E}} I_{ij}(\mu_{ij}) \right) \tag{B.14c}$$

$$\overset{(B.12f)}{=} -\log \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)} - 1. \tag{B.14d}$$

Finally, setting temporarily $\varepsilon = 1$, the corresponding Lagrangian of (B.1) reads

$$\begin{aligned} L(\mu, \nu) = \langle \theta, \mu \rangle - H(\mu) &+ \sum_{i \in \mathcal{V}} \nu_i \big(1 - \sum_{x_i \in \mathcal{X}} \mu_i(x_i)\big) \\ &+ \sum_{ij \in \mathcal{E}} \sum_{x_i \in \mathcal{X}} \nu_{ij}(x_i) \big(\sum_{x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) - \mu_i \big) \\ &+ \sum_{ij \in \mathcal{E}} \sum_{x_j \in \mathcal{X}} \nu_{ij}(x_j) \big(\sum_{x_i \in \mathcal{X}} \mu_{ij}(x_i, x_j) - \mu_j \big), \end{aligned} \tag{B.15}$$

and its partial derivatives with respect to μ are given by

$$\frac{d}{d\mu_i(x_i)} L(\mu, \nu) = \theta_i(x_i) + \log \mu_i(x_i) - \nu_i(x_i) - \sum_{j \in \mathcal{N}(i)} \nu_{ij}(x_i), \tag{B.16a}$$

$$\frac{d}{d\mu_{ij}(x_i, x_j)} L(\mu, \nu) = \theta_{ij}(x_i, x_j) + \log \frac{\mu_{ij}(x_i, x_j)}{\mu_i(x_i)\mu_j(x_j)} + 1 + \nu_{ij}(x_i) + \nu_{ij}(x_j). \tag{B.16b}$$

Evaluating the optimality condition $\nabla_\mu L(\mu, \nu) = 0$ and solving for $\log \mu_i$ and $\log \mu_{ij}$ respectively, we have

$$\log \mu_i(x_i) = -\theta_i(x_i) + \nu_i(x_i) + \sum_{j \in \mathcal{N}(i)} \nu_{ij}(x_i) \tag{B.17a}$$

$$\log \mu_{ij}(x_i, x_j) = -\theta_{ij}(x_i, x_j) + \log \mu_i(x_i) + \log \mu_j(x_j) - 1 - v_{ij}(x_i) - v_{ij}(x_j), \tag{B.17b}$$

and by inserting $\log \mu_i, \log \mu_j$ from (B.17a) into (B.17b) gives

$$\begin{aligned}\log \mu_{ij}(x_i, x_j) = &-\theta_{ij}(x_i, x_j) - \theta_i(x_i) - \theta_j(x_j) + v_i(x_i) + v_j(x_j) \\ &+ \sum_{k \in \mathcal{N}(i) \setminus \{j\}} v_{ik}(x_i) + \sum_{k \in \mathcal{N}(j) \setminus \{i\}} v_{jk}(x_j) - 1\end{aligned} \tag{B.17c}$$

Finally, the primal variables μ and the dual variables v are connected by the optimality conditions

$$\mu_i(x_i) \overset{(B.17b)}{=} e^{v_i} e^{-\theta_i(x_i)} \prod_{j \in \mathcal{N}(i)} e^{v_{ij}(x_i)}, \qquad x_i \in \mathcal{X},\ i \in \mathcal{V}, \tag{B.18a}$$

$$\mu_{ij}(x_i, x_j) \overset{(B.17c)}{=} e^{v_i + v_j} e^{-\theta_{ij}(x_i, x_j) - \theta_i(x_i) - \theta_j(x_j)} \prod_{k \in \mathcal{N}(i) \setminus \{j\}} e^{v_{ik}(x_i)} \prod_{k \in \mathcal{N}(j) \setminus \{i\}} e^{v_{jk}(x_j)}, \tag{B.18b}$$

for $x_i, x_j \in \mathcal{X},\ ij \in \mathcal{E}$. Note, that the constant e^{-1} in (B.18b) is absorbed in the dual variables v_i, v_j, with slight abuse of notation. Equations (B.18a) and (B.18b) establish (2.49a) and (2.49b), respectively.

Assume μ is an optimal primal variable, then we can eliminate this variable by marginalization

$$\begin{aligned}\sum_{x_j \in \mathcal{X}} \mu_{ij}(x_i, x_j) &\overset{(B.18b)}{=} e^{v_i} e^{-\theta_i(x_i)} \prod_{k \in \mathcal{N}(i) \setminus \{j\}} e^{v_{ik}(x_i)} \\ &\qquad \sum_{x_j \in \mathcal{X}} \left(e^{v_j} e^{-\theta_{ij}(x_i, x_j) - \theta_j(x_j)} \prod_{k \in \mathcal{N}(j) \setminus \{i\}} e^{v_{jk}(x_j)} \right),\end{aligned} \tag{B.19a}$$

$$\overset{(2.39)}{=} \mu_i(x_i) \overset{(B.18a)}{=} e^{v_i} e^{-\theta_i(x_i)} \prod_{k \in \mathcal{N}(i)} e^{v_{ik}(x_i)}, \tag{B.19b}$$

and obtain a fixed point equation only in terms of the dual variables by solving (B.19b) for v_{ij}

$$e^{v_{ij}(x_i)} = e^{v_j} \sum_{x_j \in \mathcal{X}} \left(e^{-\theta_{ij}(x_i, x_j) - \theta_j(x_j)} \prod_{k \in \mathcal{N}(j) \setminus \{i\}} e^{v_{jk}(x_j)} \right), \quad ij \in \mathcal{E}, \quad x_i \in \mathcal{X}. \tag{B.20}$$

This equation establishes *loopy belief propagation* by *message passing* given by formula (2.50) and sketched in Section 2.2.5.

Bibliography

[1] A. Aaron, J. Fakcharoenphol, C. Harrelson, R. Krauthgamer, K. Talwar, and E. Tardos. Approximate Classification via Earthmover Metrics. In *Proc. SODA*, pages 1079–1087, 2004.

[2] P.-A. Absil, R. Mahony, and R. Sepulchre. *Optimization Algorithms on Matrix Manifolds*. Princeton University Press, Princeton, NJ, 2008. ISBN 978-0-691-13298-3.

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

[4] L. Ambrosio. Variational Problems in SBV and Image Segmentation. *Acta Applicandae Mathematica*, 17(1):1–40, 1989.

[5] B. Andres, T. Beier, and J. Kappes. OpenGM: A C++ Library for Discrete Graphical Models. *CoRR*, abs/1206.0111, 2012.

[6] F. Åström, R. Hühnerbein, F. Savarino, J. Recknagel, and C. Schnörr. *MAP Image Labeling Using Wasserstein Messages and Geometric Assignment*, pages 373–385. Springer International Publishing, Cham, 2017. ISBN 978-3-319-58771-4. DOI: 10.1007/978-3-319-58771-4_30.

[7] F. Åström, R. Hühnerbein, F. Savarino, J. Recknagel, and C. Schnörr. MAP Image Labeling Using Wasserstein Messages and Geometric Assignment. In *Proc. SSVM*, LCNS 10302. Springer, 2017.

[8] F. Åström, S. Petra, B. Schmitzer, and C. Schnörr. Image Labeling by Assignment. *J. Math. Imag. Vision*, 58(2):211–238, 2017.

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

[10] H. H. Bauschke and J. M. Borwein. Legendre Functions and the Method of Random Bregman Projections. *J. Convex Analysis*, 4(1):27–67, 1997.

[11] R. Bergmann and D. Tenbrinck. A Graph Framework for Manifold-Valued Data. *SIAM Journal on Imaging Sciences*, 11(1):325–360, 2018.

[12] A. Bertozzi and A. Flenner. Diffuse Interface Models on Graphs for Classification of High Dimensional Data. *Multiscale Modeling & Simulation*, 10(3):1090–1118, 2012.

[13] S. Boyd and L. Vandenberghe. *Convex Optimization*. Cambridge Univ. Press, 7th edition, 2009.

[14] Y. Boykov and V. Kolmogorov. An Experimental Comparison of Min-Cut/Max-Flow Algorithms for Energy Minimization in Vision. *IEEE Trans. Patt. Anal. Mach. !ntell.*, 26(9):1124–1137, 2004.

[15] Y. Boykov, O. Veksler, and R. Zabih. Fast Approximate Energy Minimization via Graph Cuts. *IEEE Trans. Patt. Anal. Mach. Intell.*, 23(11):1222–1239, 2001.

[16] R. Brualdi. *Combinatorial Matrix Classes*. Cambridge Univ. Press, 2006.

[17] Y. Cao, S. Li, L. Petzold, and R. Serban. Adjoint Sensitivity Analysis for Differential-Algebraic Equations: The Adjoint DAE System and Its Numerical Solution. *SIAM Journal on Scientific Computing*, 24(3):1076–1089, Jan. 2003. ISSN 1064-8275, 1095-7197. DOI: 10.1137/S1064827501380630.

[18] Y. Censor and S. Zenios. Proximal Minimization Algorithm with *D*-Functions. *J. Optim. Theory Appl.*, 73(3):451–464, 1992.

[19] C. Chekuri, S. Khanna, J. Naor, and L. Zosin. Linear Programming Formulation and Approximation Algorithms for the Metric Labeling Problem. *SIAM J. Discr. Math.*, 18(3):608–625, 2005.

[20] M. Cuturi. Sinkhorn Distances: Lightspeed Computation of Optimal Transport. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 2292–2300. Curran Associates, Inc., 2013.

[21] M. Cuturi and G. Peyré. A Smoothed Dual Approach for Variational Wasserstein Problems. *SIAM J. Imag. Sci.*, 9(1):320–343, Jan 2016. ISSN 1936-4954.

[22] J. Danskin. The Theory of Max-Min, with Applications. *SIAM J. Appl. Math.*, 1966.

[23] M. Deza and M. Laurent. *Geometry of Cuts and Metrics*. Springer, 1997.

[24] W. E. A Proposal on Machine Learning via Dynamical Systems. *Comm. Math. and Statistics*, 5(1):1–11, 2017.

[25] A. Elmoataz, O. Lezoray, and S. Bougleux. Nonlocal Discrete Regularization on Weighted Graphs: A Framework for Image and Manifold Processing. *IEEE Trans. Image Proc.*, 17(7):1047–1059, 2008.

[26] G. Gilboa and S. Osher. Nonlocal Operators with Applications to Image Processing. *Multiscale Model. Simul.*, 7(3):1005–1028, 2008.

[27] E. Haber and L. Ruthotto. Stable Architectures for Deep Neural Networks. *Inverse Problems*, 34(1):014004, 2017.

[28] W. W. Hager. Runge-Kutta methods in optimal control and the transformed adjoint system. *Numerische Mathematik*, 87(2):247–282, 2000.

[29] E. Hairer and G. Wanner. *Solving Ordinary Differential Equations II: Stiff and Differential-Algebraic Problems*. Springer Series in Computational Mathematics, Springer Ser.Comp.Mathem. Hairer,E.:Solving Ordinary Diff. Springer-Verlag, Berlin Heidelberg, 2 edition, 1996. ISBN 978-3-540-60452-5.

[30] E. Hairer, S. P. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I: Nonstiff Problems*. Springer Series in Computational Mathematics, Springer Ser.Comp.Mathem. Hairer,E.:Solving Ordinary Diff. Springer-Verlag, Berlin Heidelberg, 2 edition, 1993. ISBN 978-3-540-56670-0.

[31] E. Hairer, C. Lubich, and G. Wanner. *Geometric Numerical Integration: Structure-Preserving Algorithms for Ordinary Differential Equations*. Springer, 2 edition, 2006.

[32] E. Hairer, S. Nørsett, and G. Wanner. *Solving Ordinary Differential Equations I*. Springer, 3rd edition, 2008.

[33] T. Hazan and A. Shashua. Norm-Product Belief Propagation: Primal-Dual Message-Passing for Approximate Inference. *IEEE Trans. Inf. Theory*, 56(12): 6294–6316, 2010.

[34] K. He, X. Zhang, S. Ren, and J. Sun. Deep Residual Learning for Image Recognition. In *Proc. CVPR*, 2016.

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

[36] R. Hühnerbein, F. Savarino, F. Åström, and C. Schnörr. Image Labeling Based on Graphical Models Using Wasserstein Messages and Geometric Assignment. *SIAM Journal on Imaging Sciences*, 11(2):1317–1362, Jan. 2018. ISSN 1936-4954. DOI: 10.1137/17M1150669.

[37] R. Hühnerbein, F. Savarino, S. Petra, and C. Schnörr. Learning Adaptive Regularization for Image Labeling Using Geometric Assignment. *arXiv:1910.09976 [cs, math]*, Oct. 2019. arXiv: 1910.09976.

[38] J. Kappes, B. Andres, F. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *Int. J. Comp. Vision*, 115(2):155–184, 2015.

[39] J. H. Kappes, B. Andres, F. A. Hamprecht, C. Schnörr, S. Nowozin, D. Batra, S. Kim, B. X. Kausler, T. Kröger, J. Lellmann, N. Komodakis, B. Savchynskyy, and C. Rother. A Comparative Study of Modern Inference Techniques for Structured Discrete Energy Minimization Problems. *arXiv:1404.0533 [cs]*, Apr. 2014. arXiv: 1404.0533.

[40] J. Kleinberg and E. Tardos. Approximation Algorithms for Classification Problems with Pairwise Relationships: Metric Labeling and Markov Random Fields. *Journal of the ACM*, 49(5):616–639, Sep 2002. ISSN 0004-5411.

[41] P. Knight. The Sinkhorn-Knopp Algorithm: Convergence and Applications. *SIAM J. Matrix Anal. Appl.*, 30(1):261–275, 2008.

[42] V. Kolmogorov. Convergent Tree-Reweighted Message Passing for Energy Minimization. *IEEE Trans. Patt. Anal. Mach. Intell.*, 28(10):1568–1583, 2006.

[43] V. Kolmogorov and R. Zabih. What Energy Functions Can Be Minimized via Graph Cuts? *IEEE Trans. Patt. Analysis Mach. Intell.*, 26(2):147–159, 2004.

[44] S. Kolouri, S. Park, M. Thorpe, D. Slepcev, and G. Rohde. Optimal mass transport: Signal processing and machine- learning applications. *IEEE Signal Proc. Mag.*, 34(4):43–59, 2017.

[45] J. Lee. *Introduction to Smooth Manifolds*. Graduate Texts in Mathematics. Springer-Verlag, New York, 2 edition, 2012. ISBN 978-1-4419-9981-8.

[46] J. Lee. *Introduction to Riemannian Manifolds*. Graduate Texts in Mathematics. Springer International Publishing, 2 edition, 2018. ISBN 978-3-319-91754-2. DOI: 10.1007/978-3-319-91755-9.

[47] J. M. Lee. *Riemannian Manifolds: An Introduction to Curvature*. Graduate Texts in Mathematics. Springer-Verlag, New York, 1997. ISBN 978-0-387-98271-7. DOI: 10.1007/b98852.

[48] A. Nemirovsky and D. Yudin. *Problem Complexity and Method Efficiency in Optimization*. John Wiley & Sons, 1983.

[49] Y. Nesterov and A. Nemirovskii. *Interior Point Polynomial Algorithms in Convex Programming*. Studies in Applied Mathematics. Society for Industrial and Applied Mathematics, 1987. ISBN 9780898715156.

[50] M. Padberg. The Boolean Quadratic Polytope: Some Characteristics, Facets and Relatives. *Math. Progr.*, 45:139–172, 1989.

[51] P. Pakzad and V. Anantharam. Estimation and Marginalization using Kikuchi Approximation Methods. *Neural Computation*, 17(8):1836–1873, 2005.

[52] G. Peyré. Entropic Approximation of Wasserstein Gradient Flows. *SIAM J. Imag. Sci.*, 8(4):2323–2351, Jan 2015. ISSN 1936-4954.

[53] T. Pham Dinh and L. Hoai An. Convex Analysis Approach to D.C. Programming: Theory, Algorithms and Applications. *Acta Math. Vietnamica*, 22(1):289–355, 1997.

[54] T. Pham Dinh and L. Hoai An. A D.C. Optimization Algorithm for Solving the Trust-Region Subproblem. *SIAM J. Optimization*, 8(2):476–505, 1998.

[55] J. Phillips. Coresets and Sketches. In *Handbook of Discrete and Computational Geometry*, chapter 48. CRC Press, 2016.

[56] P. Ravikumar, A. Agarwal, and M. J. Wainwright. Message-Passing for Graph-Structured Linear Programs: Proximal Methods and Rounding Schemes. *J. Mach. Learning Res.*, 11:1043–1080, 2010.

[57] J. Renegar. Linear Programming, Complexity Theory and Elementary Functional Analysis. *Math. Progr.*, 70:279–351, 1995.

[58] R. Rockafellar. On a Special Class of Functions. *J. Opt. Theory Appl.*, 70(3):619–621, 1991.

[59] R. Rockafellar, M. Wets, and R. Wets. *Variational Analysis*. Grundlehren der mathematischen Wissenschaften. Springer Berlin Heidelberg, 2009. ISBN 978-3-540-62772-2.

[60] R. T. Rockafellar. Monotone Operators and the Proximal Point Algorithm. *SIAM J. Control Optim.*, 14(5):877–898, 1976.

[61] I. M. Ross. A Roadmap for Optimal Control: The Right Way to Commute. *Annals of the New York Academy of Sciences*, 1065(1):210–231, 2018/09/12 2006.

[62] J. Sanz-Serna. Symplectic Runge–Kutta Schemes for Adjoint Equations, Automatic Differentiation, Optimal Control, and More. *SIAM Review*, 58(1):3–33, 2016.

[63] F. Savarino and C. Schnörr. Continuous-Domain Assignment Flows. *arXiv:1910.07287 [nlin]*, Oct. 2019. arXiv: 1910.07287.

[64] F. Savarino and C. Schnörr. A Variational Perspective on the Assignment Flow. In J. Lellmann, M. Burger, and J. Modersitzki, editors, *Scale Space and Variational Methods in Computer Vision*, Lecture Notes in Computer Science, pages 547–558, Cham, 2019. Springer International Publishing. ISBN 978-3-030-22368-7. DOI: 10.1007/978-3-030-22368-7_43.

[65] F. Savarino, R. Hühnerbein, F. Åström, J. Recknagel, and C. Schnörr. *Numerical Integration of Riemannian Gradient Flows for Image Labeling*, pages 361–372. Springer International Publishing, Cham, 2017. ISBN 978-3-319-58771-4. DOI: 10.1007/978-3-319-58771-4_29.

[66] B. Schmitzer. A Sparse Multiscale Algorithm for Dense Optimal Transport. *J. Math. Imag. Vision*, 56(2):238–259, 2016.

[67] B. Schmitzer. Stabilized Sparse Scaling Algorithms for Entropy Regularized Transport Problems. *SIAM Journal on Scientific Computing*, 41(3):A1443–A1481, Jan. 2019. ISSN 1064-8275, 1095-7197. DOI: 10.1137/16M1106018.

[68] M. Schneider. Matrix Scaling, Entropy Minimization, and Conjugate Duality (II): The Dual Problem. *Math. Progr.*, 48:103–124, 1990.

[69] C. Schnörr. Assignment Flows. In P. Grohs, M. Holler, and A. Weinmann, editors, *Variational Methods for Nonlinear Geometric Data and Applications*. Springer (in press), 2020.

[70] R. Sinkhorn. A Relationship Between Arbitrary Positive Matrices and Doubly Stochastic Matrices. *Ann. Math. Statist.*, 35(2):876–879, 06 1964.

[71] P. Swoboda, A. Shekhovtsov, J. Kappes, C. Schnörr, and B. Savchynskyy. Partial Optimality by Pruning for MAP-Inference with General Graphical Models. *IEEE Trans. Patt. Anal. Mach. Intell.*, 38(7):1370–1382, 2016.

[72] T. Terlaky. *Interior Point Methods of Mathematical Programming*. Kluwer Acad. Publ., 1996.

[73] R. Tomović and M. Vukobratovíc. *General Sensitivity Theory*. Modern Analytic and Computational Methods in Science and Mathematics. American Elsevier Pub. Co., 1972. ISBN 9780444001085.

[74] M. Wainwright and M. Jordan. Graphical Models, Exponential Families, and Variational Inference. *Found. Trends Mach. Learning*, 1(1-2):1–305, 2008.

[75] M. Wainwright, T. Jaakola, and A. Willsky. MAP Estimation via Agreement on Trees: Message-Passing and Linear Programming. *IEEE Trans. Inform. Theory*, 51(11):3697– 3717, 2005.

[76] Y. Weiss. Comparing the Mean Field Method and Belief Propagation for Approximate Inference in MRFs. In *Advanced Mean Field Methods: Theory and Practice*, pages 229–240. MIT Press, 2001.

[77] T. Werner. A Linear Programming Approach to Max-sum Problem: A Review. *IEEE Trans. Patt. Anal. Mach. Intell.*, 29(7):1165–1179, 2007.

[78] T. Werner. A Linear Programming Approach to Max-Sum Problem: A Review. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(7):1165–1179, July 2007. ISSN 0162-8828. DOI: 10.1109/TPAMI.2007.1036.

[79] C. Yanover, T. Meltzer, and Y. Weiss. Linear Programming Relaxations and Belief Propagation - An Empirical Study. *J. Mach. Learning Res.*, 7:1887–1907, 2006.

[80] J. Yedidia, W. Freeman, and Y. Weiss. Constructing Free-Energy Approximations and Generalized Belief Propagation Algorithms. *IEEE Trans. Information Theory*, 51(7):2282–2312, 2005.

[81] A. Zeilmann, F. Savarino, S. Petra, and C. Schnörr. Geometric Numerical Integration of the Assignment Flow. *Inverse Problems, in press: https://doi.org/10.1088/1361-6420/ab2772 (preprint CoRR abs/1810.06970)*, 2020.

[82] M. Zisler, A. Zern, S. Petra, and C. Schnörr. Self-Assignment Flows for Unsupervised Data Labeling on Graphs. *arXiv:1911.03472 [cs]*, Nov. 2019. arXiv: 1911.03472.

[83] M. Zisler, A. Zern, S. Petra, and C. Schnörr. Unsupervised Labeling by Geometric and Spatially Regularized Self-assignment. In J. Lellmann, M. Burger, and J. Modersitzki, editors, *Scale Space and Variational Methods in Computer Vision*, Lecture Notes in Computer Science, pages 432–444, Cham, 2019. Springer International Publishing. ISBN 978-3-030-22368-7. DOI: 10.1007/978-3-030-22368-7_34.

Nomenclature

List of Symbols

$\triangle$	End of a definition, remark or assumption
$\square$	End of a proof
I_n	Identity matrix in $\mathbb{R}^{n\times n}$
$[n]$	Set of natural numbers, $[n] = \{1, 2, \ldots, n\}$
$\operatorname{diag} v$	Diagonal matrix with the entries of the vector v on the diagonal

Blackboard Symbols

0_n	Zero Element of $\mathbb{R}^n$
$\mathbb{1}_n$	Vector of length n with all components equal to 1
$\mathbb{N}$	Set of natural numbers excluding zero
$\mathbb{R}$	Set of real numbers
$\mathbb{R}^n$	Space n-vectors with elements from $\mathbb{R}$
$\mathbb{R}^{m\times n}$	Space of $(m \times n)$-matrices with elements from $\mathbb{R}$
$\mathbb{R}^n_+$	Positive orthant, $\mathbb{R}^n_+ = \{p \in \mathbb{R}^n \colon\ p \geq 0\}$
$\mathbb{R}^n_{++}$	Set of strictly positive vectors, $\mathbb{R}^n_{++} = \{p \in \mathbb{R}^n \colon\ p > 0\}$

Calligraphic Symbols

$\mathcal{A}$	Linear map extracting *marginals* of a double stochastic matrix
$\mathcal{A}^\top$	Transposed mapping of $\mathcal{A}$
$\mathcal{E}$	Set of edges of a given graph $\mathcal{G}$
$\mathcal{F}$	Feature space
$\mathcal{G}$	Graph
$\mathcal{L}$	Local polytope
$\mathcal{N}_i$	Local neighborhood of pixel $i \in \mathcal{V}$
$\mathcal{N}$	Scaled sphere, $\mathcal{N} = 2\mathbb{S}^{n-1}$
$\mathcal{P}$	Parameter manifold, $\mathcal{P} = \mathcal{S} \times \ldots \times \mathcal{S}$
$\mathcal{X}$	Prototypes/Labels
$\mathcal{S}$	Relative interior of probability simplex Δ_{n-1}
$\mathcal{V}$	Set of vertices of a given graph $\mathcal{G}$
$\mathcal{W}$	Assignment manifold, $\mathcal{W} = \mathcal{S} \times \ldots \times \mathcal{S}$

Greek Symbols

α	Rounding parameter of $f_{\tau,\alpha}$
Δ_{n-1}	Probability simplex, $\Delta_{n-1} = \{p \in \mathbb{R}^n_+ \colon \langle \mathbb{1}_n, p \rangle = 1\}$
$\mathring{\Delta}_{n-1}$	Relative interior of probability simplex, $\mathring{\Delta}_{n-1} = \Delta_{n-1} \cap \mathbb{R}^n_{++}$
λ	Adjoint state
ω_i	Weight vector consisting of $\lvert\mathcal{N}_i\rvert$ elements with $\omega_i \in \mathring{\Delta}_{\lvert\mathcal{N}_i\rvert}$
ρ	Scaling parameter for distance matrix D
τ	Smoothing parameter of Wasserstein distance

Roman Symbols

$d_{\mathcal{F}}$	Distance function in feature space $\mathcal{F}$
D	Distance matrix
L	Likelihood matrix
S	Similarity matrix
W	Assignment matrix
V	Tangent space matrix

List of Figures

1.1 Image labeling example 1

2.1 Butcher tableau of a Runge–Kutta method 28
2.2 Computing adjoint sensitivities 30

3.1 Sphere map for the 2-dimensional simplex $\mathcal{S}_2$. 39
3.2 First-order approximation of geodesic $\gamma_v(t)$ 41
3.3 Geometry of the probability simplex induced by the Fisher–Rao metric 43
3.4 Overview of the geometric approach [8] 49
3.5 Parameter influence of the *scaling parameter* ρ and the *spatial scale* $|\mathcal{N}|$ on the assignment 54
3.6 Averaged entropy (3.55) for different values of ρ and $|\mathcal{N}|$ 55

4.1 Approximations of the indicator function $\delta_{\mathbb{R}_-}$ 67
4.2 Entropy-regularized Wasserstein distance $d_{\Theta,\tau}(c,\gamma(t))$ for varying parameter τ and increasing numbers n of labels 69
4.3 Wasserstein distance with Potts prior (4.4) on probability simplex Δ_3 . 70
4.4 Influence of the rounding parameter α and the smoothing parameter τ 77
4.5 Normalized average entropy (3.55) and smoothed energy E_τ (4.7) . . . 79
4.6 Connection between the objective function $f_{\tau,\alpha}$ (4.42) and the discrete energy E (4.2) of the underlying graphical model 80
4.7 The minimal binary cyclic graphical model $\mathcal{K}^3$ 81
4.8 Evaluation of the minimal cyclic graphical model $\mathcal{K}^3$ 84
4.9 Comparison to other methods 87
4.10 Non-Potts prior example 89

5.1 Curvilinear line structures: Training data 100
5.2 Curvilinear line structures: Input data and local label assignments . . 101
5.3 Curvilinear line structures: Training phase: Labeling results 102
5.4 Curvilinear line structures: Training phase: Optimal weight patches . 103
5.5 Coreset visualization 106
5.6 Curvilinear line structures: Test phase: Labeling results 107
5.7 Curvilinear line structures: Test phase: Predicted weight patches . . . 109
5.8 Pattern completion 110
5.9 Transporting and enlarging label assignments 111
5.10 Parameter learning vs. optimal control 112

List of Tables

2.1 Symplectic PRK coefficients induced by the explicit Euler method . . 33
2.2 Symplectic PRK coefficients induced by Heun's method 34

4.1 Minimal cyclic graphical model on $\mathcal{K}^3$: Solutions of the marginal polytope $\mathcal{M}_{\mathcal{K}^3}$, the local polytope $\mathcal{L}_{\mathcal{K}^3}$ and our method 83
4.2 Minimal cyclic graphical model on $\mathcal{K}^3$: Three different parameter configurations and corresponding results 85

List of Acronyms

BP	Belief Propagation
BVP	Boundary Value Problem
IVP	Initial Value Problem
ODE	Ordinary Differential Equation
PRK	Partitioned Runge–Kutta
RK	Runge–Kutta
e.g.	for example (*exempli gratia*)
et al.	and others (*et alii*)
i.e.	that is (*id est*)
l.h.s.	left hand side
r.h.s.	right hand side
s.t.	subject to
w.r.t.	with respect to