Convex Optimization for Multi-Class Image Labeling with a Novel Family of Total Variation Based Regularizers

J. Lellmann, F. Becker and C. Schnörr
Image and Pattern Analysis & HCI
Dept. of Mathematics and Computer Science, University of Heidelberg
{lellmann, becker, schnoerr}@math.uni-heidelberg.de

Abstract

We introduce a linearly weighted variant of the total variation for vector fields in order to formulate regularizers for multi-class labeling problems with non-trivial inter-class distances. We characterize the possible distances, show that Euclidean distances can be exactly represented, and review some methods to approximate non-Euclidean distances in order to define novel total variation based regularizers. We show that the convex relaxed problem can be efficiently optimized to a prescribed accuracy with optimality certificates using Nesterov’s method, and evaluate and compare our approach on several synthetical and real-world examples.

1. Introduction

1.1. Overview and Motivation

The multi-class image labeling problem consists in finding, for each pixel \( x \) in the image domain \( \Omega \subseteq \mathbb{R}^d \), a label \( \ell(x) \in \{1, \ldots, l\} \) which assigns one of \( l \) class labels to \( x \) so that the labeling function \( \ell \) adheres to some data fidelity as well as spatial coherency constraints.

We consider a partial linearization of this combinatorial problem: Identify label \( i \) with the \( i \)-th unit vector \( e^i \in \mathbb{R}^l \), set \( E := \{e^1, \ldots, e^l\} \), and find

\[
\inf_{u: \Omega \rightarrow E} f(u), \quad f(u) := \int_{\Omega} \langle u(x), s(x) \rangle dx + J(u), \tag{1}
\]

The data term assigns to each label \( u(x) = e^i \) a local cost \( s_i(x) \), while the regularizer \( J \) enforces the desired spatial coherency. In terms of Markov Random Fields, the data and regularization terms can be thought of as unary and pairwise potentials, respectively. To tackle the combinatorial nature of (1), the problem is relaxed,

\[
\inf_{u \in C} \int_{\Omega} \langle u(x), s(x) \rangle dx + J(u), \tag{2}
\]

where \( C := \{u : \Omega \rightarrow \mathbb{R}^l | u_i(x) \geq 0, \sum_{i=1}^l u_i(x) = 1\} \) is a convex set which constrains each \( u(x) \) to the unit simplex.

As the data term was linearized, the local costs \( s \) may be arbitrarily complex, possibly derived from a probabilistic model, without affecting the overall problem class. In particular, any convex regularizer \( J \) yields a continuous convex problem, which can be globally optimized. It is thus interesting to examine the expressiveness of convex regularizers. In this paper, we study a class of convex total variation (TV) based regularizers,

\[
J(u) = \int_{\Omega} \|D(Au)\|_F \, dx, \tag{3}
\]
where \( \|D(\cdot)\|_F \) is the Frobenius norm of the Jacobian (in a distributional sense), and \( A \in \mathbb{R}^{k \times l} \) is a weights matrix that will be used to vary the regularization cost according to a distance \( d(i, j) \) between the labels of the adjoining regions.

The motivation comes from the fact that the classical total variation regularizer advocates discrete solutions, and thus produces a unique labeling. In the classical two-class formulation, the total variation penalizes adjacent, differently labeled regions according to the area of the interface. Our formulation carries over this property to the multi-class case in a precise sense, and allows for more general, non-uniform distances. In particular, Euclidean distances can be represented exactly.

With a suitable discretization, the problem corresponds to a bilinear saddle point problem, which we propose to solve using a method suggested by Nesterov. The method does not require to set any parameters other than the desired optimality, and provides \( \varepsilon \)-optimal solutions in \( O(1/\varepsilon) \).

1.2. Related Work

The continuous two-class case – optimization on the set of characteristic functions – is known as continuous cut [19]. Chan et al. [6] showed that this problem can be solved on a relaxed, convex set without losing global optimality.

For anisotropic discretization, the binary case can be formulated as a minimum-cut problem on a grid graph, which allows to solve the problem exactly and efficiently for a large class of metrics using graph cuts [12, 3]. This formulation and its anisotropic multi-class generalization can be viewed as pairwise binary respective multilabel Markov Random Fields (MRFs). Prominent methods to handle the multi-class case rely on finding a local minimum by solving a sequence of binary graph cuts [4] (see [13] for a recent generalization), while we solve one convex problem to a global optimum.

Our results can be seen as a continuous analogon to [9], where it was shown that convex pairwise energies of a special form can be exactly formulated as a cut on a multi-layered graph. An early analysis can be found in [11], where the authors also derive suboptimality bounds of a linear programming relaxation for metric distances. All these methods rely on the graph representation with pairwise potentials, while our approach has its roots in the continuous setting. While this complicates optimization due to the higher order potentials, it avoids the metrization error of the discrete methods and permits fast parallel optimization.

In the continuous setting, closely related to our approach is [5]. The authors use a linearization as in [9], and give a thorough analysis of the continuous model where \( d \) is of the form \( \sigma(|i - j|) \) for nondecreasing, positive, concave \( \sigma \). They propose a convexification based on the convex envelope, which gives almost discrete solutions in many cases. However their optimization method requires the selection of a step size, does not necessarily converge and requires expensive iterative projections in each iteration. Also, due to the implicit representation of their regularizer, evaluating the objective becomes a problem. In contrast, our method has guaranteed convergence and requires only simple and exactly computable projections.

Our approach is a generalization of [14] and [23], where the same linearization is used with the regularizer restricted to the Potts distance, and with less strong convergence results. An analysis of Nesterov’s method in the context of \( \ell_1 \)-norm and TV minimization can be found in [21].

1.3. Contribution

The main contribution will be twofold:

- We formulate requirements on the regularizer \( J \) and show their implications on the choice of the distance \( d \). We study the continuous formulation of the regularizer (3), and show that Euclidean distances can be represented exactly in a well-defined way (section 2, Prop. 2). We also review some methods for the approximation of non-Euclidean distances (section 2.4).
- We study the discretization of (2) in a very general saddle point formulation, and show that any specific instance can be optimized by a method suggested by Nesterov. The method is virtually parameter-free and provides explicit a priori and a posteriori optimality bounds (section 3).

In contrast to existing graph-based methods, we provide a continuous and isotropic formulation for a restricted set of distances \( d \), while in comparison with existing continuous approaches, we provide a generalization for non-uniform distances and completely characterize the convergence properties of our optimization method.

Finally, we illustrate and compare our method with the primal-dual technique from [5] and demonstrate its applicability on real-world problems (section 4).

1.4. Notation

The image domain \( \Omega \subseteq \mathbb{R}^d \) is a bounded, open, connected subset with piecewise smooth boundary \( \partial \Omega \). Superscripts \( v^k \) denote a collection of vectors, while subscripts \( v_k \) denote vector components. We denote by \( \Delta_l := \{ x \in \mathbb{R}^l | x \geq 0, e^\top x = 1 \} \) the unit simplex in \( \mathbb{R}^l \), where \( e := (1, \ldots, 1) \in \mathbb{R}^l \). \( I_n \) is the identity matrix in \( \mathbb{R}^n \) and \( \| \cdot \| \) the usual (Euclidean) 2-norm. \( B_r(x) \) denotes the ball of radius \( r \) in \( x \), and \( \chi_S(x) \) the characteristic function of \( S \).

2. Convex Functionals for Metric Labeling

We begin by formalizing the requirements on the regularizer as sketched in the introduction. Let us assume we
are given a general distance mapping \(d : \{1, \ldots, l\}^2 \to \mathbb{R}\). We do not assume any metric properties (i.e. symmetry or triangle inequality) for now. For \(u \in \mathcal{C}\), we postulate that the regularizer should satisfy

(P1) \(J\) is convex and positively homogeneous on the relaxed set \(\mathcal{C}\) as defined after (2).

(P2) \(J(u) = 0\) for any constant \(u\), i.e. there is no penalty for constant labelings.

(P3) For any partition \((S, S^c)\) of \(\Omega\) into two sets with finite perimeter \(\text{Per}(S) < \infty\), and any \(i, j \in \{1, \ldots, l\}\),

\[
J(e^i \chi_S + e^j \chi_{S^c}) = d(i, j) \text{Per}(S) .
\]

That is, a change from label \(i\) to label \(j\) gets penalized proportional to \(d(i, j)\) as well as the perimeter of the interface.

Requirements (P3) and (P2) formalize the principle that the multilabeling problem should reduce to the continuous cut in the two-class case, while the convexity from (P1) together with the linear data term renders global optimization tractable. Positive homogeneity is included as it allows \(J\) to be represented as a support function (i.e. its convex conjugate is an indicator function), which will be exploited by our optimization method. Together, these requirements pose a natural restriction on \(d\):

Proposition 1 If \((J, d)\) satisfy (P1) – (P3), it follows that \(d\) satisfies, for all \(i, j, k \in \{1, \ldots, l\}\),

1. \(d(i, i) = 0\),
2. \(d(i, j) = d(j, i) \geq 0\),
3. \(d(i, k) \leq d(i, j) + d(j, k)\).

Thus, if \(d(i, j) \neq 0\) for all \(i \neq j\), \(d\) must be a metric.

Proof see Appendix.

Consequently, for non-metric \(d\), we generally cannot expect to find such a regularizer, independent of the representation. Even for metric \(d\), existence of such a \(J\) is not clear. However it turns out that for the special case of \(d\) being an Euclidean distance, such a regularizer always exists. In the following sections, we will derive this regularizer using a total variation based approach and demonstrate how approximation methods for non-Euclidean distances can be used to still obtain an overall convex functional.

2.1. A Novel Family of TV-Based Regularizers

The classical definition for the total variation of a scalar-valued function \(u \in L^1(\Omega)\) is

\[
\text{TV}_c(u) := \sigma_{\text{div}} \mathcal{D}_c(u) := \sup_{v \in \mathcal{D}_c} \int_{\Omega} u \text{div} v \, dx ,
\]

where \(\mathcal{D}_c := \{v = (v_1, \ldots, v_d) \in (C^1_c)^d \|v(x)\|_2 \leq 1 \forall x \in \Omega\}, C^1_c \subseteq L^1\) is the space of continuously differentiable functions with compact support in \(\Omega\), and \(\sigma\) is the support function from convex analysis [17]. This formulation can be extended to vector-valued \(u \in (L^1(\Omega))^l\) by setting

\[
\mathcal{D}_v := \{v \in (C^1_c)^{l \times d} \|v(x)\|_F \leq 1 \forall x \in \Omega\},
\]

\[
\text{TV}_v(u) := \sigma_{\text{div}} \mathcal{D}_v(u) := \sup_{v \in \mathcal{D}_v} \int_{\Omega} \langle u, \text{div} v \rangle \, dx ,
\]

with \(\text{div} v := (\text{div} v^1, \ldots, \text{div} v^l)\).

Denote by \(\text{BV}(\Omega, \mathbb{R}^l)\) the functions \(u \in L^1(\Omega, \mathbb{R}^l)\) with \(\text{TV}_v(u) < \infty\), and by \(\text{Per}(S) := \text{TV}_c(\chi_S)\) the perimeter of \(S\). If \(u\) has a weak derivative \(Du\) (i.e. the Jacobian if \(u\) is continuously differentiable), we get the more instructive

\[
\text{TV}_v(u) = \int_{\Omega} \|Du\|_F \, dx ,
\]

where \(\| \cdot \|_F\) is the Frobenius norm on \(\mathbb{R}^{l \times d}\). This “classical” definition has also been used in color denoising and is sometimes referred to as MTV [18, 7]. We propose to extend this definition for our purpose by choosing an embedding matrix \(A \in \mathbb{R}^{k \times l}\), and defining

\[
\text{TV}_A(u) := \text{TV}_v(Au) .
\]

For sufficiently smooth \(u\), \(\text{TV}_A\) conforms to (3). The rest of this paper will focus on properties and applications of (9).

2.2. Properties of the Regularizer

\(\text{TV}_A\) is clearly isotropic, and convex as the composition of a convex functional and a linear operator. To further clarify the definition, let us again assume \(u\) has a weak derivative \(Du\). Then we may rewrite (9) to

\[
\text{TV}_A(u) = \int_{\Omega} \sqrt{\|D_1 u\|^2_A + \ldots + \|D_d u\|^2_A} ,
\]

where \(\|v\|_A := (v^T A^T Au)^{1/2}\). The key observation is the following, which allows to reduce \(\text{TV}_A\) to the classical total variation on a one-dimensional subspace of \(C\):

Proposition 2 Let \(a \in \mathbb{R}^l, 0 \neq b \in \mathbb{R}^l\) and \(u \in \text{BV}(\Omega, \mathbb{R})\), i.e. \(\text{TV}_c(u) < \infty\). Then

\[
\text{TV}_v(a + ub) = \|b\| \text{TV}_c(u) .
\]

Proof see Appendix.

Corollary 1 Let \(a, b \in \mathbb{R}^l\) and \(S \subseteq \Omega\) with \(\text{Per}(S) < \infty\). Then

\[
\text{TV}_v(a \chi_S + b \chi_{S^c}) = \|b-a\| \text{TV}_c(\chi_S) = \|b-a\| \text{Per}(S) .
\]
That is, interfaces of perimeter (i.e. length or area) \( \text{Per}(S) \) between two constant regions of \( u \) contribute \( \text{Per}(S) \| b-a \| \) to the overall regularization term. In particular, for \( A = (a^1 \cdots a^l) \in \mathbb{R}^{k \times l} \), we have

\[
TV_A e^i x_S + e^j x_S^c = \| a^i - a^j \| \text{Per}(S) .
\] (11)

As a byproduct, \( TV_A \) reduces nicely to the usual total variation for the two-class case:

**Corollary 2** Let \( u' \in BV(\Omega, \mathbb{R}) \) and \( A := (1/\sqrt{2}) I_2 \). Then

\[
TV_A \left( (u', 1-u')^T \right) = TV_c(u') .
\] (12)

We can thus convert any instance of the classical binary “continuous cut” approach [6] with data \( s' \in L^1(\Omega) \),

\[
\min_{u': \Omega \to [0,1]} \langle u', s' \rangle + \lambda TV_c(u)
\] (13)
to the multi-class approach (and vice versa) by the special choice of \( A := (1/\sqrt{2}) I_2 \), \( u = e^i u' + e^j (1-u') \) and assuring \( s_1 - s_2 = s' \), e.g. \( s = e^i s' \). This will be considerably generalized in the following section.

### 2.3. Exact Representation of Euclidean Distances

We will first look at Euclidean distances \( d \), i.e. there is a \( k \in \mathbb{N} \) and \( x^1, \ldots, x^l \in \mathbb{R}^k \) with \( d(i,j) = \| x^i - x^j \| \). Then we have the following result:

**Proposition 3** Let \( d \) be an Euclidean distance. Then there exist \( k \in \mathbb{N} \) and \( A \in \mathbb{R}^{k \times l} \) s.t. \( J(u) := TV_A(u) \) satisfies (P1)–(P3).

**Proof** Comparing (11) to (4), we see that we may just use the embedding matrix \( A = (x^1 \cdots x^l) \).

The class of Euclidean distances comprises some important special cases:

- The uniform, discrete or Potts distance as also considered in [14, 23] and as a special case in [11, 13], \( d(i,j) = 0 \) iff \( i = j \) and \( d(i,j) = 1 \) in any other case, when \( A = (1/2) I \).

- The linear (label) distance, \( d(i,j) = |i-j| \), with \( A = (e, 2e, \ldots, le) \). This regularizer is suitable to problems where the labels can be naturally ordered, e.g. depth from stereo or grayscale image denoising.

- More generally, if label \( i \) corresponds to a prototypical vector \( x^i \) in \( k \)-dimensional feature space, and the Euclidean norm is an appropriate metric on the features, it is natural to set \( d(i,j) = \| x^i - x^j \| \), which is Euclidean by construction. This corresponds to a regularization in feature space, rather than in “label space”.

Non-metric or non-Euclidean \( d \), such as the truncated label distance, \( d(i,j) = \min\{2, |i-j| \} \), cannot be represented exactly by \( TV_A \). Yet, tight approximations preserving convexity of the overall problem exist, as shown next.

### 2.4. Approximation for General Distances

Now let \( d \) be a general metric with squared matrix representation \( D \in \mathbb{R}^{l \times l} \), \( D_{ij} = d(i,j)^2 \). Then it is known [2, Chap. 12] that \( d \) is Euclidean iff for \( C := I - \frac{1}{2} ee^T \), the matrix \( T := -\frac{1}{2} CDC \) is positive semidefinite. In this case, \( A \) can be found by factorizing \( T = A^T A \). If \( T \) is not positive semidefinite, dropping the nonnegative eigenvalues in \( T \) yields an Euclidean approximation. This method known as classical scaling [2] does not necessarily give good absolute error bounds.

For non-metric, nonnegative \( d \), we can formulate the problem of finding the “closest” Euclidean distance matrix \( E \) as minimization of a matrix norm \( \| E - D \|_M \) over all \( E \in E_1 \), the set of \( l \times l \) Euclidean distance matrices. Fortunately, there is a linear bijection \( B : P_{l-1} \rightarrow E_1 \) between \( E_1 \) and the space of positive semidefinite \( (l-1) \times (l-1) \) matrices \( P_{l-1} \) [8, 10]. This allows us to rewrite our problem as a semidefinite program [22, p.534–541],

\[
\inf_{S \in P_{l-1}} \| B(S) - D \|_M .
\] (14)

The resulting problem can be solved using available numerical solvers. Then \( E = B(S) \in E_1 \), and \( A \) can be extracted by factorizing \( -\frac{1}{2} CEC \). Since \( E \) and \( D \) are explicitly known, we can compute an a posteriori bound on the maximum distance error, \( \varepsilon_E := \max_{i,j} |E_{ij} - D_{ij}| \). Fig. 2 shows a visualization of some embeddings of for a four-class problem. As an illustration, for the non-Euclidean truncated label distance, the original respective approximated distance matrices were

\[
\begin{pmatrix}
0 & 1 & 1.5 & 1.92 & 2.08 \\
1 & 0 & 1 & 1.5 & 1.92 \\
2 & 1 & 0 & 1.92 & 1.15 \\
2 & 2 & 1 & 0 & 1.15 \\
\end{pmatrix}
\]

with an absolute error bound of \( \varepsilon_E = 0.145 \).

Based on the embedding matrices computed in this way, the variational problem (2), (3) enables us to approximate the labeling problem by convex optimization for a considerably larger class of distances between labels.

### 3. Discrete Problem and Optimization

We now return to solving the discretization of (2) with the regularizer \( J = TV_A \). We will show that the discretized
In analogy to (6), we define the convex sets to compute optimality bound on the primal objective, and then provide an algorithm for optimizing (15).

3.1. Discretization of the TV-based Regularizer

If this implies strong duality, i.e. \[ \max_{v \in \mathcal{D}} f_d(v) = f_d(u^*), \]

If \( f_d \) and \( f \) can be explicitly computed, any \( v \in \mathcal{D} \) gives an optimality bound on the primal objective,

In our case, \( \mathcal{C}, \mathcal{D} \) exhibit a product structure, which allows to compute \( f \) and \( f_d \) as well as their orthogonal projections \( \Pi_\mathcal{C} \) and \( \Pi_\mathcal{D} \) efficiently, a fact that will prove important in the algorithmic part. We first show that the segmentation problem (2) belongs to this class under a suitable discretization, and then provide an algorithm for optimizing (15).

3.2. Discretization of the TV-based Regularizer

We discretize \( \Omega \) by a regular grid, \( \Omega = \{1, \ldots, n_1\} \times \ldots \times \{1, \ldots, n_d\} \subseteq \mathbb{R}^d, d \in \mathbb{N} \), consisting of \( n := |\Omega| \) pixels, and \( u \) by its (vectorial) values at the pixels in \( \Omega \), i.e. \( u \in \mathbb{R}^{n \times l} \). The multidimensional image space \( X := \mathbb{R}^{n \times l} \) is equipped with the Euclidean inner product \( \langle \cdot, \cdot \rangle \) over the vectorized elements. We naturally identify \( v = (v^1, \ldots, v^l) \in \mathbb{R}^{n \times l} \) with \( ((v^1)\top \ldots (v^l)\top)\top \in \mathbb{R}^{nl} \).

Let \( \text{grad} := (\text{grad}_{1}^{\top}, \ldots, \text{grad}_{l}^{\top})^{\top} \) be the \( d \)-dimensional forward difference gradient operator for Neumann boundary conditions. Accordingly, \( \text{div} := -\text{grad}^{\top} \) is the backward difference divergence operator for Dirichlet boundary conditions. These operators extend to \( \mathbb{R}^{n \times l} \) via \( \text{Grad} := (I_l \otimes \text{grad}), \text{Div} := (I_l \otimes \text{div}) \), where \( I_l \) is the \( l \times l \) identity matrix. In analogy to (6), we define the convex sets

\[ C := \{ u \in \mathbb{R}^{n \times l} | u_{i,\cdot} \in \Delta_l, i = 1, \ldots, n \}, \]

\[ D_{\text{loc}} := \{ p = (p^1, \ldots, p^l) \in \mathbb{R}^{d \times l} | \|p\|_1 \leq 1 \}, \]

\[ D := \prod_{x \in \Omega} D_{\text{loc}} \subseteq \mathbb{R}^{n \times d \times l}. \]

The discrete total variation on vector-valued data is then

\[ \text{TV}_V(u) := \sigma_{\text{Div}}(u) = \sigma_{\text{Div}}(\text{Grad} u) = \sum_{x \in \Omega} \|G_x u\|_2, \]

where \( G_x \) is an \((ld) \times n\) matrix composed of rows of \( (\text{Grad}) s.t. G_x u \) gives the gradients of all \( u_x \) in \( x \) stacked one above the other. By modifying \( D_{\text{loc}} \), we could easily replace the Euclidean norm by e.g. the 1-norm. For \( A \) as in (9), define

\[ \text{TV}_A(u) := \text{TV}_V((A \otimes I_n)u) = \sigma_{\text{D}}(Lu) \]

with \( L := (\text{Grad})(A \otimes I_n) \).

We finally arrive at the form (15) with \( \mathcal{C}, \mathcal{D} \), and \( L \) defined as above, \( m = nk \) and \( b = 0 \). Projections on \( \mathcal{C} \) and \( \mathcal{D} \) are highly separable and thus can be computed easily (cf. [15]). The same holds for the primal and dual objectives \( f \) and \( f_d \): the former using (20), the latter is just a sum of vector minimum functions. Note that in contrast to the continuous framework, we may easily substitute non-homogeneous, spatially varying or even nonlocal regularizers by choosing \( L \) appropriately.

3.2. Nesterov optimization

To optimize (15), we follow the work of Nesterov [16]. The algorithm has a theoretical complexity bound of \( O(1/\varepsilon) \) for finding an \( \varepsilon \)-optimal solution, and has been shown to give accurate results for denoising [1] and general \( \ell_1 \)-norm based problems [21]. The complete algorithm for our saddle point formulation is shown in Alg. 1. The only expensive operations are the projections \( \Pi_\mathcal{C} \) and \( \Pi_\mathcal{D} \), which are efficiently computable as shown above. The algorithm converges in the objective in any case and provides an explicit optimality certificate:

---

Algorithm 1: Convex Multi-Class Labeling

1. input: \( c_1 \in \mathcal{C}, c_2 \in \mathcal{D} \) and \( r_1, r_2 \in \mathbb{R} \) s.t. \( \mathcal{C} \subseteq \mathcal{B}_{r_1}(c_1) \) and \( \mathcal{D} \subseteq \mathcal{B}_{r_2}(c_2) \); \( x^{(0)} \in \mathcal{C} \); \( x \in \mathcal{C} \); \( N \in \mathbb{N} \).
2. output: \( u^{(N)} \in \mathcal{C}, v^{(N)} \in \mathcal{D} \).
3. let \( \mu := \frac{2\|L\|}{N+1} r_2 \).
4. set \( G^{(-1)} = 0, v^{(-1)} = 0 \).
5. for \( k = 0, \ldots, N \) do
6. \( V \leftarrow \Pi_{\mathcal{D}} \left( c_2 + \frac{1}{N} (L x^{(k)} - b) \right) \).
7. \( u^{(k)} \leftarrow v^{(k-1)} + 2 \frac{G^{(k-1)} + \frac{1}{2} \mu G}{N+1} V \).
8. \( G \leftarrow s + L^{\top} V \).
9. \( G^{(k)} \leftarrow G^{(k-1)} + \frac{1}{2} \mu G \).
10. \( u^{(k)} \leftarrow \Pi_{\mathcal{C}} \left( x^{(k)} - \frac{\mu}{\|L\|} G \right) \).
11. \( z^{(k)} \leftarrow \Pi_{\mathcal{C}} \left( c_1 - \frac{\mu}{\|L\|} G^{(k)} \right) \).
12. \( x^{(k+1)} \leftarrow \frac{2}{k+\frac{1}{k+\gamma}} z^{(k)} + \left( 1 - \frac{2}{k+\gamma} \right) u^{(k)} \).
13. end for
Proposition 4 For a solution $u^*$ of (15), it holds that $u^{(N)} \in C$, $v^{(N)} \in D$, i.e. $u^{(N)}$ and $v^{(N)}$ are primal resp. dual feasible, and

$$f(u^{(N)}) - f(u^*) \leq f(u^{(N)}) - f_d(v^{(N)}) \leq \frac{2r_1 r_2 C}{(N+1)}, \quad (22)$$

Proof Apply [16, Thm. 3] with $\tilde{f}(u) = \langle u, s \rangle$, $A = L$, $\phi(v) = \langle b, v \rangle$, $d_1(u) := \frac{1}{2} \|u - c_1\|^2$, $d_2(v) := \frac{1}{2} \|v - c_2\|^2$, $D_1 = \frac{1}{2} r_1^2$, $D_2 = \frac{1}{2} r_2^2$, $\sigma_1 = \sigma_2 = 1$, $M = 0$. □

Corollary 3 For given $\varepsilon > 0$, applying Alg. 1 with

$$N = \left[ 2r_1 r_2 C \varepsilon^{-1} - 1 \right]$$

yields an $\varepsilon$-optimal solution to (15).

For our discretization (21), we may choose $c_1 = \frac{1}{\sqrt{N}}$, $r_1 = \sqrt{n(l-1)/l}$, $c_2 = 0$, $r_2 = \sqrt{n}$, and $C = \sqrt{2d\|A\|} \geq \|\text{Grad} \|A\| \geq \|L\|$. We arrive at a parameter-free algorithm with a total complexity of $O(\varepsilon^{-1} n \sqrt{d\|A\|})$ iterations to find $u^{(N)}$ with a suboptimality of at most $\varepsilon$, i.e. $f(u^{(N)}) - f(u^*) \leq \varepsilon$. Note that this allows us to solve any problem of the saddle point form (15), which allows for many generalizations such as spatially varying distances, alternate linearization schemes, as long as the projections $\Pi_C$ and $\Pi_D$ can be computed.

3.3. Optimality

After solving the relaxed problem, a binary solution, i.e. a hard labeling, needs to be recovered. For the continuous two-class case, [6] showed that an exact solution can be obtained by thresholding at almost any threshold. However, their results do not immediately transfer to the discrete multi-class case. In particular, the crucial coarea formula holds for $\ell_1$-, but not $\ell_2$-discretizations of the TV.

There seems to be no obvious “best” choice for the binarization scheme. As taking the index of first maximal component does not work well for degenerate (i.e. rank deficient) $A$, we chose the final class label for pixel $x^i$ as the smallest index $j$ minimizing $\|e^j - u(x^i)\|_A$, which worked well in all considered cases. While we are not aware of an a priori bound on the error introduced by binarization in this case, (17) and (22) provide an a posteriori optimality bound: As the binary approximate solution $\bar{u}^{(N)}$ is primal feasible,

$$f(\bar{u}^{(N)}) - f(\bar{u}^*) \leq f(u^{(N)}) - f_d(v^{(N)}). \quad (24)$$

Computation of both $f$ and $f_d$ is efficient for the discretization as outlined above.

4. Experimental Results

Fig. 3 visualizes the effect of varying $d$ on the segmentation. The foreground-background ($fg$-$bg$) distance was chosen to clearly segment the three foreground classes from the white background, while allowing for a high variance between foreground classes. We found that for highly degenerate embeddings $A$, as is the case for the linear distance, our approach tends to generate non-binary solutions. This results in a large energy increase during binarization (Table 4); however the returned lower bound allows us to detect these cases. In all experiments, we clamped the pixel values to $[0,1]$, which results in a significant amount of salt-and-pepper – despite originally Gaussian – noise. In all segmentation experiments, the data term is a simple $\ell_1$-distance to the prototypical color vectors for each class.

<table>
<thead>
<tr>
<th>Distance</th>
<th>Potts</th>
<th>fg-bg</th>
<th>linear</th>
</tr>
</thead>
<tbody>
<tr>
<td>$f(u^{(N)})$</td>
<td>15898.3</td>
<td>14942.0</td>
<td>13797.28</td>
</tr>
<tr>
<td>$f(\bar{u}^{(N)})$</td>
<td>16006.1</td>
<td>15047.50</td>
<td>16860.82</td>
</tr>
<tr>
<td>$f_d(v^{(N)})$</td>
<td>15879.3</td>
<td>14919.15</td>
<td>13783.44</td>
</tr>
<tr>
<td>rel. gap relaxed</td>
<td>0.0012</td>
<td>0.0015</td>
<td>0.0010</td>
</tr>
<tr>
<td>rel. gap discrete</td>
<td>0.0080</td>
<td>0.0086</td>
<td>0.2233</td>
</tr>
</tbody>
</table>
pending on the actual labels, we may introduce classes.

optimization of non-Euclidean distances: The truncated linear distance (3.98% incorrectly labeled). The latter is non-Euclidean and approximated using the method outlined in section 2.4. This result shows that an accurate non-binary image labeling can be obtained by solving a single convex optimization problem.

Figure 5. Stereo disparity estimation with non-Euclidean distance using the data term from [20]. Each pixel is assigned one of 16 disparity labels. 

Left: Input image. Second from left: Ground truth. Second from right: Potts distance (5.75% incorrectly labeled). Right: Truncated linear distance (3.98% incorrectly labeled). The latter is non-Euclidean and approximated using the method outlined in section 2.4. This result shows that an accurate non-binary image labeling can be obtained by solving a single convex optimization problem.

Figure 6. Objective vs. number of iterations $N$ using the Arrow-Hurwicz method from [5] (dashed) with various choices for the step size $\tau$ and our method (solid) on the problem in Fig. 5. The proposed parameter-free Nesterov method shows competitive performance, and outperforms the Arrow-Hurwicz method if the step size is not hand-tuned to the specific dataset.

In Fig. 4, we compare the region-filling properties of our method to a standard four-neighborhood approach optimized using alpha-beta-swap. The algorithm converges to a clearly non-binary result; however after binarization we get the correct minimal partition as expected from the continuous formulation. Direct numerical comparison to the graph cut methods is difficult, as the latter rely on binary potentials, while our discretization uses ternary potentials.

Fig. 5 demonstrates the applicability of our method for optimization of non-Euclidean distances: The truncated linear distance improves stereo disparity estimation, as it allows for small depth variations within objects and does not excessively penalize depth changes at object borders.

To characterize the performance of our optimization method, we compared it to the Arrow-Hurwicz method from [5] with alternating primal and dual proximal steps, applied to our relaxation. The latter method requires to set a step size, which leads to divergence if set too large. We found the upper bound to be dependent on the embedding $A$. On the Tsukuba data set (Fig. 5), both methods are close, but the convergence speed of the Arrow-Hurwicz method depends on the manually chosen step size (Fig. 6). We also found that our method generally outperforms the theoretical suboptimality bound (22) by a factor of $5 - 10$.

Since our approach allows to vary the regularization depending on the actual labels, we may introduce classes which differ only by the regularization term. In Fig. 7, the grayscale image was segmented into dark and light background, as well as two foreground classes for text over dark and text over light background with identical data term. By choosing a suitable (originally non-Euclidean) $d$, this allows to simultaneously separate foreground from background and perform a background reconstruction, which is not possible using the standard Potts distance.

5. Conclusion and Future Work

Based on a total variation for vector fields, we showed how to formulate a class of convex spatial regularizers in the continuous multi-labeling framework. The regularizers may depend on arbitrary distances between labels, which are approximated if necessary. Solving the discrete optimization problem using Nesterov’s method showed to be competitive in speed without requiring any parameter tuning.

The promising results should motivate to investigate what other kinds of regularizers are possible. The optimization framework allows for many other – possibly higher-order, nonlocal, or spatially varying – regularizers and different relaxations by replacing the constraint sets $C$ and $D$, with the only requirement that projections on these sets can be computed.
References


6. Appendix

Proof of Proposition 1. 1. follows from (P2) and (P3) by choosing i = j and S with Per(S) > 0. Symmetry in 2. is obtained from (P2) by replacing S with S′, as Per(S) = Per(S′). For 3., fix any S with c := Per(S) > 0, then cd(i, k) = cJ(e′χS + e′′χS′) ≤ cJ(e′χS + e′′χS′) + J(e′χS′ + e′′χS) = c(d(i, j) + d(j, k)) due to (P1). Thus we may substitute ⟨b, v⟩ := ∥b∥v′ into (P1) and (P2).

Proof of Proposition 2. Rearranging the terms and applying Gauss’ theorem to remove the constant part yields

\[ TV_c (a + ub) = \sup_{v \in D_c} \int_\Omega u \diver (\langle b, v_1 \rangle, \ldots, \langle b, v_d \rangle) dx. \]

Now for any v ∈ D_c, there exists v′ ∈ D_c s.t. \( \forall i \in \{1, \ldots, d\} : \langle b, v_i \rangle = ∥b∥v′_i \) and vice versa: For v ∈ D_c, set v′_i := ∥b∥^-1\langle b, v_i \rangle, then ∥v′(x)∥^2 ≤ 1. For v′ ∈ D_c set v_i(x) := ∥b∥^-1b\langle v′_i(x) \rangle, then ∥v(x)∥_F = ∥v′(x)∥_2 ≤ 1. Thus we may substitute ⟨b, v⟩ = ∥b∥v′, and

\[ TV_c (a + ub) = \sup_{v \in D_c} \int_\Omega u ∥b∥ v′ dx = ∥b∥ TV_c(u). \]