The Total Variation on Hypergraphs - Learning on Hypergraphs Revisited

M. Hein, S. Setzer, L. Jost, S. Rangapuram

Preprint 153
Edited by

AG Numerik/Optimierung
Fachbereich 12 - Mathematik und Informatik
Philipps-Universität Marburg
Hans-Meerwein-Str.
35032 Marburg
The Total Variation on Hypergraphs - Learning on Hypergraphs Revisited

M. Hein, S. Setzer, L. Jost, S. Rangapuram

Preprint 153
The consecutive numbering of the publications is determined by their chronological order.

The aim of this preprint series is to make new research rapidly available for scientific discussion. Therefore, the responsibility for the contents is solely due to the authors. The publications will be distributed by the authors.
The Total Variation on Hypergraphs - Learning on Hypergraphs Revisited

Matthias Hein, Simon Setzer, Leonardo Jost and Syama Sundar Rangapuram
Department of Computer Science
Saarland University

Abstract

Hypergraphs allow to encode higher-order relationships in data and are thus a very flexible modeling tool. Current learning methods are either based on approximations of the hypergraphs via graphs or on tensor methods which are only applicable under special conditions. In this paper, we present a new learning framework on hypergraphs which fully uses the hypergraph structure. The key element is a family of regularization functionals based on the total variation on hypergraphs.

1 Introduction

Graph-based learning is by now well established in machine learning and is the standard way to deal with data that encode pairwise relationships. Hypergraphs are a natural extension of graphs which allow to model also higher-order relations in data. It has been recognized in several application areas such as computer vision [1, 2], bioinformatics [3, 4] and information retrieval [5, 6] that such higher-order relations are available and help to improve the learning performance.

Current approaches in hypergraph-based learning can be divided into two categories. The first one uses tensor methods for clustering as the higher-order extension of matrix (spectral) methods for graphs [7, 8, 9]. While tensor methods are mathematically quite appealing, they are limited to so-called $k$-uniform hypergraphs, that is, each hyperedge contains exactly $k$ vertices. Thus, they are not able to model mixed higher-order relationships. The second main approach can deal with arbitrary hypergraphs [10, 11]. The basic idea of this line of work is to approximate the hypergraph via a standard weighted graph. In a second step, one then uses methods developed for graph-based clustering and semi-supervised learning. The two main ways of approximating the hypergraph by a standard graph are the clique and the star expansion which were compared in [12]. One can summarize their results by stating that there exists no approximation which fully encodes the hypergraph structure. Earlier it has been proven that an exact representation of the hypergraph via a graph retaining its cut properties is impossible [13].

In this paper, we overcome the limitations of both existing approaches. We note that both for clustering and semi-supervised learning the key element, either explicitly or implicitly, is the cut functional. Our aim is to directly work with the cut defined on the hypergraph. We discuss in detail the differences of this hypergraph cut and the cut functionals induced by the clique and star expansion in Section 2.1. Then, in Section 2.2, we introduce the total variation on a hypergraph as the Lovasz extension of the hypergraph cut. Based on this, we propose a family of regularization functionals which interpolates between the total variation and a regularization functional enforcing smoother functions on the hypergraph corresponding to Laplacian-type regularization on graphs. We show a model for semi-supervised learning based on these regularizers in Section 3. Furthermore, in the line of recent research on tight relaxations of balanced graph cuts [14, 15, 16, 17], we show in Section 4 that there exists a tight relaxation of the normalized hypergraph cut. In both applications, a new type of convex optimization problems has to be solved for which we derive first-order gradient methods in Section 5. The main ingredients of these algorithms are proximal mappings for which
we provide a novel algorithm and analyze its complexity. In the experimental section 6, we show that fully incorporating hypergraph structure is beneficial in semi-supervised learning and clustering on hypergraphs. All proofs are moved to the supplementary material.

2 The Total Variation on Hypergraphs

A large class of graph-based algorithms in semi-supervised learning and clustering is based either explicitly or implicitly on the cut. Thus, we discuss first in Section 2.1 the hypergraph cut and the corresponding approximations of [11, 12]. In Section 2.2, we introduce in analogy to the total variation on graphs, the total variation on hypergraphs as the Lovasz extension of the hypergraph cut as well as a corresponding family of regularization functionals.

2.1 Hypergraphs, Graphs and Cuts

Hypergraphs allow to model relations which are not only pairwise as in graphs but involve multiple vertices. In this paper, we consider weighted undirected hypergraphs $H = (V, E, w)$ where $V$ is the vertex set with $|V| = n$ and $E$ the set of hyperedges with $|E| = m$. Each hyperedge $e \in E$ corresponds to a subset of vertices, i.e., to an element of $2^V$. The vector $w \in \mathbb{R}^m$ contains for each hyperedge $e$ its non-negative weight $w_e$. In the following, we use the letter $H$ also for the incidence matrix $H \in \mathbb{R}^{|V| \times |E|}$ which is for $i \in V$ and $e \in E$,

$$H_{i,e} = \begin{cases} 1 & \text{if } i \in e, \\ 0 & \text{else}. \end{cases}$$

The degree of a vertex $i \in V$ is defined as $d_i = \sum_{e \in E} w_e H_{i,e}$ and the cardinality of an edge $e$ can be written as $|e| = \sum_{i \in V} H_{j,e}$. We would like to emphasize that we do not impose the restriction that the hypergraph is $k$-uniform, i.e., that each hyperedge contains exactly $k$ vertices.

The considered class of hypergraphs contains the set of undirected, weighted graphs which is equivalent to the set of 2-uniform hypergraphs. The motivation for the total variation on hypergraphs comes from the correspondence between the cut on a graph and the total variation functional. Thus, we recall the definition of the cut on weighted graphs $G = (V, W)$ with weight matrix $W$. Let $\overline{C} = V \setminus C$ denote the complement of $C$ in $V$. Then, for a partition $(C, \overline{C})$, the cut is defined as

$$\text{cut}_C(C, \overline{C}) = \sum_{i,j : i \in C, j \in \overline{C}} w_{ij}.$$  

This standard definition of the cut carries over naturally to a hypergraph $H$

$$\text{cut}_H(C, \overline{C}) = \sum_{e \in C \neq \emptyset, \; e \cap \overline{C} \neq \emptyset} w_e.$$  

Thus, the cut functional on a hypergraph is just the sum of the weights of the hyperedges which have vertices both in $C$ and $\overline{C}$. It is not biased towards a particular way the hyperedge is cut, that is, how many vertices of the hyperedge are in $C$ resp. $\overline{C}$. This emphasizes that the vertices in a hyperedge belong together and we penalize every cut of a hyperedge with the same value.

In order to handle hypergraphs with existing methods developed for graphs, the focus in previous works [11, 12] has been on transforming the hypergraph into a graph. In [11], they suggest to use the clique expansion (CE), i.e., every hyperedge $e \in H$ is replaced with a fully connected subgraph where every edge in this subgraph has weight $\frac{w_e}{|e|}$. This leads to the cut functional $\text{cut}_{CE}$.

$$\text{cut}_{CE}(C, \overline{C}) = \sum_{e \in C \neq \emptyset, \; e \cap \overline{C} \neq \emptyset} \frac{w_e}{|e|} |e \cap C| |e \cap \overline{C}|.$$  

Note that in contrast to the hypergraph cut (1), the value of $\text{cut}_{CE}$ depends on the way each hyperedge is cut since the term $|e \cap C| |e \cap \overline{C}|$ makes the weights dependent on the partition. In particular, the smallest weight is attained if only a single vertex is split off, whereas the largest weight is attained if the partition of the hyperedge is most balanced. In comparison to the hypergraph cut, this leads to a bias towards cuts that favor to split off single vertices from a hyperedge which in our point of view is an undesired property for most applications. We illustrate this with an example in Figure 1, where the minimum hypergraph cut (cut$_H$) leads to a balanced partition, whereas the minimum
In this section, we define the total variation on hypergraphs. The key technical element is the Lovasz extension which extends a set function, seen as a mapping on $\mathcal{P}(V)$.

### 2.2 The Total Variation on Hypergraphs

![Minimum hypergraph cut](image)

**Proposition 2.1.** Suppose $H = (V, E, w)$ is a weighted 3-uniform hypergraph. Then, $W \in \mathbb{R}^{V \times V}$ defined as $W = \frac{1}{2}H \text{diag}(w)H^T$ defines the weight matrix of a graph $G = (V, W)$ where each cut of $G$ has the same value as the corresponding hypergraph cut of $H$.

**Proof.** The cut value of a partition $(C, \overline{C})$ of $G$ is given as

$$\text{cut}_G(C, \overline{C}) = \frac{1}{2} \sum_{e \in E} |e \cap C||e \cap \overline{C}|w_e.$$  

The product $|e \cap C||e \cap \overline{C}|$ takes the values $2$ if $e$ is cut by $C$ and zero otherwise. Because of the factor $\frac{1}{2}$, we thus get equivalence to the hypergraph cut. \qed

**Definition 2.1.** Let $\hat{S} : 2^V \to \mathbb{R}$ be a set function with $\hat{S}(\emptyset) = 0$. Let $f \in \mathbb{R}^V$, let $V$ be ordered such that $f_1 \leq f_2 \leq \ldots \leq f_n$ and define $C_i = \{j \in V \mid j > i \}$. Then, the **Lovasz extension** $S : \mathbb{R}^V \to \mathbb{R}$ of $\hat{S}$ is given by

$$S(f) = \sum_{i=1}^n f_i (\hat{S}(C_{i-1}) - \hat{S}(C_i)) = \sum_{i=1}^{n-1} \hat{S}(C_i)(f_{i+1} - f_i) + f_1 \hat{S}(V).$$

Note that for the characteristic function of a set $C \subset V$, we have $S(1_C) = \hat{S}(C)$.

It is well-known that the Lovasz extension $S$ is a convex function if and only if $\hat{S}$ is submodular [18]. For graphs $G = (V, W)$, the total variation on graphs is defined as the Lovasz extension of the graph cut [18] given as $\text{TV}_G : \mathbb{R}^V \to \mathbb{R}$, $\text{TV}_G(f) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} |f_i - f_j|$.

**Proposition 2.2.** The total variation $\text{TV}_H : \mathbb{R}^V \to \mathbb{R}$ on a hypergraph $H = (V, E, w)$ defined as the Lovasz extension of the hypergraph cut, $\hat{S}(C) = \text{cut}_H(C, \overline{C})$, is a convex function given by

$$\text{TV}_H(f) = \sum_{e \in E} w_e \left( \max_{v \in e} f_i - \min_{j \in e} f_j \right) = \sum_{e \in E} w_e \max_{i,j \in e} |f_i - f_j|.$$  

Finally, note that for weighted 3-uniform hypergraphs it is always possible to find a corresponding graph such that any cut of the graph is equal to the corresponding cut of the hypergraph.

**Proposition 2.2.** The total variation $\text{TV} : \mathbb{R}^V \to \mathbb{R}$ on a hypergraph $H = (V, E, w)$ defined as

$$\text{TV}(f) = \sum_{e \in E} w_e \left( \max_{v \in e} f_i - \min_{j \in e} f_j \right) = \sum_{e \in E} w_e \max_{i,j \in e} |f_i - f_j|,$$  

The clique expansion cut $(\text{cut}_{CE})$ not only cuts an additional hyperedge but is also unbalanced. This is due to its bias towards splitting off single nodes of a hyperedge. Another argument against the clique expansion is computational complexity. For large hyperedges the clique expansion leads to (almost) fully connected graphs which makes computations slow and is prohibitive for large hypergraphs.

We omit the discussion of the star graph approximation of hypergraphs discussed in [12] as it is shown there that the star graph expansion is very similar to the clique expansion. Instead, we want to recall the result of Ihler et al. [13] which states that in general there exists no graph with the same vertex set $V$ which has for every partition $(C, \overline{C})$ the same cut value as the hypergraph cut.
Proof. Using \( C_{i-1} = C_i \cup \{i\} \) and \( \overline{C_i} = \overline{C_{i-1}} \cup \{i\} \) the Lovasz extension can be written as

\[
\text{TV}_H(f) = \sum_{i=1}^n f_i \left( \text{cut}(C_{i-1}, \overline{C_{i-1}}) - \text{cut}(C_i, \overline{C_i}) \right) = \sum_{i=1}^n f_i \left( \text{cut}(\{i\}, \overline{C_{i-1}}) - \text{cut}(C_i, \overline{C_i}) \right)
\]

\[
= \sum_{i=1}^n f_i \left( \sum_{e \in [1, \ldots, i-1] \neq \emptyset} w_e - \sum_{e \in [i, \ldots, n] \neq \emptyset} w_e \right) = \sum_{e \in E} w_e \left( \max_{i \in e} f_i - \min_{j \in e} f_j \right).
\]

It is easy to see that the Lovasz extension of the hypergraph cut is a convex function. Since the maximum of convex functions is convex, \( \min_{i \in e} f_i = \max_{i \in e} f_i \) and the hyperedge weights are non-negative, we have a non-negative combination of convex functions which is convex. Alternatively, one could use that the hypergraph cut is submodular and the Lovasz extension of every submodular set function is convex.

Note that the total variation of a hypergraph cut reduces to the total variation on graphs if \( H \) is 2-uniform (standard graph). There is an interesting relation of the total variation on hypergraphs to sparsity inducing group norms. Namely, defining for each edge \( e \in E \) the difference operator \( D_e : \mathbb{R}^{|V|} \to \mathbb{R}^{|V| \times |E|} \) by \( (D_e f)_{ij} = f_i - f_j \) if \( i, j \in e \) and 0 otherwise, \( \text{TV}_H \) can be written as, \( \text{TV}_H(f) = \sum_{e \in E} w_e \|D_e f\|_\infty \), which can be seen as inducing group sparse structure on the gradient level. The groups are the hyperedges and thus are typically overlapping. This could lead potentially to extensions of the elastic net on graphs to hypergraphs.

It is known that using the total variation on graphs as a regularization functional in semi-supervised learning (SSL) leads to very spiky solutions for small numbers of labeled points. Thus, one would like to have regularization functionals enforcing more smoothness of the solutions. For graphs this is achieved by using the family of regularization functionals \( \Omega_{G,p} : \mathbb{R}^{|V|} \to \mathbb{R} \),

\[
\Omega_{G,p}(f) = \frac{1}{2} \sum_{i,j=1}^n w_{ij} |f_i - f_j|^p.
\]

For \( p = 2 \) we get regularization functional of the (combinatorial) graph Laplacian which is the basis of a large class of methods for SSL on graphs. In analogy to graphs, we define the corresponding family of regularizers on hypergraphs.

**Definition 2.2.** The regularization functionals \( \Omega_{H,p} : \mathbb{R}^{|V|} \to \mathbb{R} \) for a hypergraph \( H = (V, E, w) \) are defined for \( p \geq 1 \) as

\[
\Omega_{H,p}(f) = \sum_{e \in E} w_e \left( \max \left\{ f_i \in e \right\} - \min \left\{ f_j \in e \right\} \right)^p.
\]

**Lemma 2.1.** The functionals \( \Omega_{H,p} : \mathbb{R}^{|V|} \to \mathbb{R} \) are convex.

**Proof.** The \( p \)-th power of positive, convex functions for \( p \geq 1 \) is convex as

\[
(f(x) + (1 - \lambda)y)^p \leq (\lambda f(x) + (1 - \lambda)f(y))^p \leq \lambda f(x)^p + (1 - \lambda)f(y)^p
\]

where the last inequality follows from the convexity of \( x^p \) on \( \mathbb{R}_+ \). Thus, the \( p \)-th power of \( \max \limits_{i \in e} f_i - \min \limits_{j \in e} f_j \) is convex. \( \Box \)

Note that \( \Omega_{H,1}(f) = \text{TV}_H(f) \). If \( H \) is a graph, \( \Omega_{H,2} \) reduces to the Laplacian regularization \( \Omega_{G,2} \).

Note that for characteristic functions of sets, \( f \equiv 1_C \), it holds \( \Omega_{H,p}(1_C) = \text{cut}_H(C, \overline{C}) \). Thus, the difference between the hypergraph cut and its approximations such as clique and star expansion carries over to \( \Omega_{H,p} \) and \( \Omega_{G,C;p} \).

### 3 Semi-supervised Learning

With the regularization functionals derived in the last section, we can immediately write down a formulation for two-class semi-supervised learning on hypergraphs similar to the well-known approaches of \[19, 20\]. Given the label set \( L \) we construct the vector \( Y \in \mathbb{R}^n \) with \( Y_i = 0 \) if \( i \notin L \)
and \( Y_i \) equal to the label in \( \{-1, 1\} \) if \( i \in L \). We propose to solve
\[
f^* = \arg\min_{f \in \mathbb{R}^{|V|}} \frac{1}{2} \| f - Y \|_2^2 + \lambda \Omega_{H,p}(f),
\]
where \( \lambda > 0 \) is the regularization parameter. In Section 5, we discuss how this convex optimization problem can be solved efficiently for the case \( p = 1 \) and \( p = 2 \). Note, that other loss functions than the squared loss could be used. However, the regularizer aims at contracting the function and we use the label set \( \{-1, 1\} \) so that \( f^* \in [-1, 1]^{|V|} \). However, on the interval \([ -1, 1 ] \) the squared loss behaves very similar to other margin-based loss functions.

In general, we recommend to use \( p = 2 \) as it corresponds to Laplacian-type regularization for graphs which is known to work well. For graphs \( p = 1 \) is known to produce spiky solutions for small numbers of labeled points. This is due to the effect that cutting “out” the labeled points leads to a much smaller cut than, e.g., producing a balanced partition. However, in the case where one has only a small number of hyperedges this effect is much smaller and we will see in the experiments that \( p = 1 \) also leads to reasonable solutions.

### 4 Balanced Hypergraph Cuts

In Section 2.1, we discussed the difference between the hypergraph cut (1) and the graph cut of the clique expansion (2) of the hypergraph and gave a simple example in Figure 1 where these cuts yield quite different results. Clearly, this difference carries over to the famous normalized cut criterion introduced in [21, 22] for clustering of graphs with applications in image segmentation. For a hypergraph the ratio resp. normalized cut can be formulated as
\[
RCut(C, \overline{C}) = \frac{\text{cut}_H(C, \overline{C})}{|C||\overline{C}|}, \quad NCut(C, \overline{C}) = \frac{\text{cut}_H(C, \overline{C})}{\text{vol}(C) \text{vol}(\overline{C})},
\]
which incorporate different balancing criteria. Note, that in contrast to the normalized cut for graphs the normalized hypergraph cut allows no relaxation into a linear eigenproblem (spectral relaxation).

Thus, we follow a recent line of research [14, 15, 16, 17] where it has been shown that the standard spectral relaxation of the normalized cut used in spectral clustering [22] is loose and that a tight relaxation can be formulated in terms of a nonlinear eigenproblem. Although nonlinear eigenproblems are non-convex, one can compute nonlinear eigenvectors quite efficiently at the price of loosing global optimality. However, it has been shown that the potentially non-optimal solutions of the tight relaxation, outperform in practice the globally optimal solution of the loose relaxation, often by large margin. In this section, we extend their approach to hypergraphs and consider general balanced hypergraph cuts \( Bcut(C, \overline{C}) \) of the form, \( Bcut(C, \overline{C}) = \frac{\text{cut}_H(C, \overline{C})}{\text{vol}(C) \text{vol}(\overline{C})} \), where \( \hat{S} : 2^V \to \mathbb{R}_+ \) is a non-negative, symmetric set function (that is \( \hat{S}(C) = \hat{S}(\overline{C}) \)). For the normalized cut one has \( \hat{S}(C) = \text{vol}(C) \text{vol}(\overline{C}) \) whereas for the Cheeger cut one has \( \hat{S}(C) = \min\{\text{vol}C, \text{vol}\overline{C}\} \). Other examples of balancing functions can be found in [16]. Our following result shows that the balanced hypergraph cut also has a tight relaxation into a continuous nonlinear eigenproblem [14].

**Theorem 4.1.** Let \( H = (V, E, w) \) be a finite, weighted hypergraph and \( S : 2^V \to \mathbb{R} \) be the Lovasz extension of the symmetric, non-negative set function \( \hat{S} : 2^V \to \mathbb{R} \). Then, it holds that
\[
\min_{f \in \mathbb{R}^{|V|}} \frac{\sum_{e \in E} w_e (\max_{i \in e} f_i - \min_{j \in e} f_j)}{S(f)} = \min_{C \subseteq V} \frac{\text{cut}_H(C, \overline{C})}{S(C)}.
\]

Further, let \( f \in \mathbb{R}^{|V|} \) and define \( C_t := \{ i \in V \mid f_i > t \} \). Then,
\[
\min_{t \in \mathbb{R}} \frac{\text{cut}_H(C_t, \overline{C_t})}{S(C_t)} \leq \frac{\sum_{e \in E} w_e (\max_{i \in e} f_i - \min_{j \in e} f_j)}{S(f)}.
\]

**Proof.** By Prop. 2.2 the Lovasz extension of \( \text{cut}_H(C, \overline{C}) \) is given by \( \sum_{e \in E} w_e (\max_{i \in e} f_i - \min_{j \in e} f_j) \).

Noting that both \( \text{cut}_H(C, \overline{C}) \) and \( \hat{S}(C) \) vanish on the full set \( V \), the proof then follows from the recent result [17], which shows in this case the equivalence between the set problem and the continuous problem written in terms of the Lovasz extensions. 

\[\square\]
The last part of the theorem shows that “optimal thresholding” among all level sets of \( f \in \mathbb{R}^{V_1} \) can only lead to a better or equally balanced hypergraph cut.

The question remains how to minimize the ratio \( Q(f) = \frac{TV_H(f)}{S(f)} \). As discussed in [16], every Lovasz extension \( S \) can be written as a difference of convex positively 1-homogeneous functions\(^1\) \( S = S_1 - S_2 \). Moreover, as shown in Prop. 2.2 the total variation \( TV_H \) is convex. Thus, we have to minimize a non-negative ratio of a convex and a d.c. function\(^2\). For this we use the RatioDCA algorithm of [16], which for completeness is shown in Algorithm 1. The main part of RatioDCA is

\begin{algorithm}
\begin{enumerate}
    \item \textbf{Input:} \( Q = \frac{R_1 - R_2}{S_1 - S_2} \)
    \item \textbf{Initialization:} \( f^0 = \text{random with } \|f^0\|_2 = 1 \), \( \lambda^0 = Q(f^0) \)
    \item \textbf{repeat}
    \item \hspace{0.5cm} \( s_1(f^k) \in \partial S_1(f^k), r_2(f^k) \in \partial R_2(f^k) \)
    \item \hspace{0.5cm} \( f^{k+1} = \text{arg} \min_{\|u\|_2 \leq 1} \{ R_1(u) - \langle u, r_2(f^k) \rangle + \lambda^k (S_2(u) - \langle u, s_1(f^k) \rangle) \} \)
    \item \hspace{0.5cm} \( \lambda^{k+1} = \frac{R_1(f^{k+1}) - R_2(f^{k+1})}{(S_1(f^{k+1}) - S_2(f^{k+1}))} \)
    \item \hspace{0.5cm} \textbf{until} \( |\lambda^{k+1} - \lambda^k| < \epsilon \)
    \item \textbf{Output:} eigenvalue \( \lambda^{k+1} \) and eigenvector \( f^{k+1} \).
\end{enumerate}
\end{algorithm}

the inner problem which is a convex optimization problem. In our case \( R_1 = TV_H, R_2 = 0 \) and the inner problem reads

\[
\min_{\|u\|_2 \leq 1} \{ TV_H(u) + \lambda^k (S_2(u) - \langle u, s_1(f^k) \rangle) \}.
\]

While the strategy for general balancing functions is discussed in [16], we restrict ourselves for simplicity to submodular balancing functions, in which case \( S \) is convex and thus \( S_2 = 0 \). Note that the balancing functions of ratio/normalized cut and Cheeger cut are submodular. It turns out that the inner problem is very similar to the semi-supervised learning formulation (3). The efficient solution of both problems is discussed in the next section.

5 Algorithms for the Total Variation on Hypergraphs

The problem (3) we want to solve for semi-supervised learning and the inner problem (4) of RatioDCA have a common structure. They are the sum of convex functionals where one of them is the novel regularizer \( \Omega_H, \rho \). We propose to solve these problems using a primal-dual algorithm, denoted PDHG in this paper, which was proposed in [23, 24]. Its main idea is to iteratively solve for each defined by \( \Omega \) a so-called proximal problem. Solving the proximal problem \( PDHG \) in this paper, which was proposed in [23, 24].

The main idea here is that often these proximal problems can be solved efficiently leading to a fast convergence of the overall algorithm. In order to point out the common structure of PDHG for both (3) and the inner problems of Algorithm 1, we first consider a general optimization problem of the form

\[
\min_{f \in \mathbb{R}^n} \{ G(f) + F(Kf) \},
\]

where \( K \in \mathbb{R}^{m,n} \) and \( G : \mathbb{R}^n \rightarrow \mathbb{R}, F : \mathbb{R}^n \rightarrow \mathbb{R} \) are lower-semicontinuous convex functions. Recall that the conjugate function of \( G^* \) of \( G \) is defined as

\[
G^*(x) = \sup_{f \in \mathbb{R}^n} \{ \langle x, f \rangle - G(f) \}
\]

and similarly for \( F^* \). In terms of these conjugate functions, we can write the dual problem of (5) as

\[
- \min_{\alpha \in \mathbb{R}} \{ G^*(-K^T \alpha) + F^*(\alpha) \}.
\]

\(^1\) A function \( f : \mathbb{R}^d \rightarrow \mathbb{R} \) is positively 1-homogeneous if \( \forall \alpha > 0 \), \( f(\alpha x) = \alpha f(x) \).

\(^2\) A function is called d.c. if it can be written as a difference of convex functions
The PDHG algorithm for (5) has the following general form. For convergence proofs we refer to [23, 24].

Algorithm 2 PDHG

1: Initialization: $f^{(0)} = f^*(0) = 0, \theta \in [0, 1], \sigma, \tau > 0$ with $\sigma\tau < 1/\|K\|_2^2$
2: repeat
3: $\alpha^{(k)} = \text{prox}_{\sigma F^*}(\alpha^{(k)} + \sigma K^2 f^{(k)})$
4: $f^{(k+1)} = \text{prox}_{\tau G^*}(f^{(k)} - \tau K^2(\alpha^{(k)}))$
5: $\bar{f}^{(k+1)} = f^{(k+1)} + \theta(\bar{f}^{(k+1)} - f^{(k)})$
6: until relative duality gap $< \epsilon$
7: Output: $f^{(k+1)}$.

We will now apply this general setting to the convex optimization problems arising in this paper. First, the following Table 1 shows how one can choose $G$ in (5) in order to solve (3) and (4), provides the solutions of the corresponding proximal problems, and gives the conjugate functions. Note that we write the constraint in the inner problem of RatioDCA via the indicator function defined by $\tau \|x\|_1(x) = 0$, if $\|x\|_2 \leq 1$ and $+\infty$ otherwise. Clearly, both proximal problems have an explicit solution.

<table>
<thead>
<tr>
<th>$G(f)$</th>
<th>$G(f) = -(s_1(f^k), f) + |x|_1(f)$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\text{prox}_{\tau G^*}(\bar{x}) = \frac{1}{\tau+\sigma}(\bar{x} + \sigma Y)$</td>
<td>$\text{prox}_{\tau G^*}(\bar{x}) = \max(1, |x|_1(\tau^k))/\tau$</td>
</tr>
<tr>
<td>$G^*(x) = \frac{1}{2}|x + Y|_2^2 - \frac{1}{2}|Y|_2^2$</td>
<td>$G^*(x) = |x + s_1(f^k)|_2^2$</td>
</tr>
</tbody>
</table>

Table 1: Data-dependent terms of the SSL functional (3) (left) and the inner problem of RatioDCA (4) (right) with respective proximal map and conjugate.

Second, we discuss the choice of $F$ and $K$ to incorporate $\Omega_{H,p}$.

PDHG algorithm for $\Omega_{H,1}$. Let $m_e$ denote the number of vertices in hyperedge $e \in E$. The main idea is to write

$$
\Omega_{H,1}(f) = F(K f) := \sum_{e \in E} (F_{(e,1)}(K_e f) + F_{(e,2)}(K_e f)),
$$

where the rows of the matrices $K_e \in \mathbb{R}^{m_e \times n}$ are the $i$-th standard unit vectors for $i \in e$ and the functionals $F_{(e,j)} : \mathbb{R}^n \rightarrow \mathbb{R}$ are defined as

$$
F_{(e,1)}(\alpha^{(e,1)}) = \lambda w_e \max(\alpha^{(e,1)}), \quad F_{(e,2)}(\alpha^{(e,2)}) = -\lambda w_e \min(\alpha^{(e,2)}).
$$

The primal problem has thus the form

$$
\min_{f \in \mathbb{R}^n} \{G(f) + \sum_{e \in E} (F_{(e,1)}(K_e f) + F_{(e,2)}(K_e f))\}.
$$

In contrast to the function $G$, we need in the PDHG algorithm the proximal maps for the conjugate functions of $F_{(e,j)}$. They are given by

$$
F_{(e,1)}^* = \ell S_{\lambda w_e}, \quad F_{(e,2)}^* = -\ell S_{\lambda w_e},
$$

where $S_{\lambda w_e} = \{x \in \mathbb{R}^n : \sum_{i=1}^{m_e} x_i = \lambda w_e, x_i \geq 0\}$ is the scaled simplex in $\mathbb{R}^n$. By (6) the dual problem has the form

$$
-\min_{\alpha^{(e,1)}, \alpha^{(e,2)}} \{G^* \left(-\sum_{e \in E} \lambda^2 K_e^2(\alpha^{(e,1)} + \alpha^{(e,2)}) + \sum_{e \in E} (\ell S_{\lambda w_e}(\alpha^{(e,1)}) + -\ell S_{\lambda w_e}(\alpha^{(e,2)})\}\}
$$

where $G^*$ is given as in Table 1. The solutions of the proximal problems for $F_{(e,1)}^*$ and $F_{(e,1)}^*$ are simply the orthogonal projections onto these simplexes written here as $P_{\ell S_{\lambda w_e}}$ and $P_{-\ell S_{\lambda w_e}}$, respectively. These projections can be performed in linear time, cf., [25].
Algorithm 3 PDHG for $\Omega_{H,1}$

1: **Initialization:** $f^{(0)} = \bar{f}^{(0)} = 0$, $\theta \in [0,1]$, $\sigma, \tau > 0$ with $\sigma \tau < 1/(2 \max_{i=1,...,n} \{c_i\})$
2: **repeat**
3: $\alpha^{(e,1)}(k+1) = P_{\mathcal{S}_{\alpha_{\infty}}}^{e} \left( \alpha^{(e,1)}(k) + \sigma K_e \bar{f}(k) \right)$, $e \in E$
4: $\alpha^{(e,2)}(k+1) = P_{-\mathcal{S}_{\alpha_{\infty}}}^{e} \left( \alpha^{(e,2)}(k) + \sigma K_e \bar{f}(k) \right)$, $e \in E$
5: $f^{(k+1)} = \text{prox}_{\tau G} \left( f^{(k)} - \tau \sum_{e \in E} K_e^T (\alpha^{(e,1)}(k+1) + \alpha^{(e,2)}(k+1)) \right)$
6: $\bar{f}^{(k+1)} = f^{(k+1)} + \theta (f^{(k+1)} - f^{(k)})$
7: **until** relative duality gap $< \epsilon$
8: **Output:** $f^{(k+1)}$

Using the proximal mappings we have presented so far, we obtain Algorithm 3. In line 1, $c_i$ is the number of hyperedges the vertex $i$ lies in. This bound on the product of the step sizes can be derived as follows

$$\|K\|^2_2 = \|K^T K\|_2 = 2 \sum_{e \in E} K_e^T K_e \|_2 = 2 \max_{i=1,...,n} \{c_i\}.$$  

It is important to point out here that the algorithm decouples the problem in the sense that in every iteration we solve subproblems which treat the functionals $G, F_{(e,1)}, F_{(e,2)}$ separately and thus can be solved in an efficient way.

**PDHG algorithm for $\Omega_{H,2}$.** We define $G$ and $K_e$ as above. Moreover, we set

$$F_e(\alpha^e) = \lambda w_e \left( \max(\alpha^e) - \min(\alpha^e) \right)^2.$$  

(8)

Hence, the primal problem can be written as

$$\min_{f \in \mathbb{R}^e} \{ G(f) + \sum_{e \in E} F_e(K_e f) \}.$$  

To formulate the dual problem we need the conjugate of $F_e$. To this end, we first derive the conjugate function of $h_e$ defined in (8), i.e.,

$$h_e^*(\alpha^e) = \sup_{\phi \in \mathbb{R}^e} \{ \langle \alpha^e, \phi \rangle - (\max(\phi) - \min(\phi))^2 \}.$$  

**Lemma 5.1.** Let $\alpha^e \in \mathbb{R}^e$ and $t_+ = \sum_{i : \alpha_i > \lambda} \alpha_i$ and $t_- = \sum_{i : \alpha_i < \lambda} \alpha_i$. It holds that

$$h_e^*(\alpha^e) = \left\{ \begin{array}{ll} \frac{1}{2} t_+^2 & \text{if } \langle \alpha^e, 1 \rangle = 0, \\ +\infty & \text{otherwise.} \end{array} \right.$$  

**Proof.** Using the decomposition, $\phi = \psi + \gamma 1$, where $\langle \psi, 1 \rangle = 0$ and $\gamma \in \mathbb{R}$, we can write

$$\langle \alpha^e, \phi \rangle - (\max(\phi) - \min(\phi))^2 = \gamma \langle \alpha^e, 1 \rangle + \langle \alpha^e, \psi \rangle - (\max(\psi) - \min(\psi))^2.$$  

Thus for $\langle \alpha^e, 1 \rangle \neq 0$, we have $h_e^*(\alpha^e) = \infty$. Now we consider the case where $\langle \alpha^e, 1 \rangle = 0$. We write $I_- = \{i : \alpha_i < \lambda\}$ and $I_+ = \{i : \alpha_i > \lambda\}$ and define $t_+ = \sum_{i \in I_+} \alpha_i$ and $t_- = \sum_{i \in I_-} \alpha_i$. Note that $\langle \alpha^e, 1 \rangle = 0$ implies $t_+ = -t_-$. Let us assume $a = \max(\phi)$ and $b = \min(\phi)$ are fixed. To maximize $\langle \alpha^e, \phi \rangle - (\max(\phi) - \min(\phi))^2$ it is clearly best to choose $\phi_i = a$ for $i \in I_-$ and $\phi_i = b$ for $i \in I_+$. Consequently,

$$\langle \alpha^e, \phi \rangle - (\max(\phi) - \min(\phi))^2 = t_+(b - a) - (b - a)^2.$$  

We maximize the gap $\Delta = b - a$ for the objective $m(\Delta) = t_+ \Delta - \Delta^2$ and obtain the maximizer as

$$\Delta = \frac{t_+}{2}.$$  

Thus we have $h_e^*(\alpha^e) = \frac{1}{4} t_+^2$ if $\langle \alpha^e, 1 \rangle \neq 0$. We consider $t_+ = \sum_{i : \alpha_i > \lambda} \alpha_i$ and $t_- = \sum_{i : \alpha_i < \lambda} \alpha_i$ we thus get

$$F_e^*(\alpha^e) = \lambda w_e h^* \left( \frac{\alpha^e}{\lambda w_e} \right) = \left\{ \begin{array}{ll} \frac{1}{4 \lambda w_e} t_+^2 & \text{if } t_+ = -t_-, \\ +\infty & \text{otherwise.} \end{array} \right.$$  

(10)
So, we obtain the dual problem
\[
- \min_{\mathbf{\alpha}} \{ G^\prime \left( - \sum_{e \in E} K_e^2 \mathbf{\alpha}^e \right) + \sum_{e \in E} \frac{1}{4\lambda} \mathbf{\alpha}^e_+ (t_e^0)^2 + \sum_{e \in E} \zeta(t_e^0 + t_e^-) \},
\]
where \( t_e^+ = \sum_{i : \mathbf{\alpha}_i^e > 0} \mathbf{\alpha}_i^e \) and \( t_e^- = \sum_{i : \mathbf{\alpha}_i^e < 0} \mathbf{\alpha}_i^e \).

As we have seen in \((10)\), the conjugate functions \( F_e^\ast \) are not indicator functions and we thus solve the corresponding proximal problems via proximal problems for \( F_e \). More specifically, we exploit the fact that
\[
\text{prox}_{\sigma F_e^\ast} (\tilde{\mathbf{\alpha}}^e) = \tilde{\mathbf{\alpha}}^e - \text{prox}_{\sigma F_e} (\tilde{\mathbf{\alpha}}^e),
\]
see \([26, \text{Lemma 2.10}]\), and use the following novel result concerning the proximal problem on the right-hand side of \((11)\).

**Proposition 5.1.** For any \( \sigma > 0 \) and any \( \tilde{\mathbf{\alpha}}^e \in \mathbb{R}^{m_e} \) the proximum
\[
\text{prox}_{\sigma F_e} (\tilde{\mathbf{\alpha}}^e) = \arg \min_{\mathbf{\alpha}^e \in \mathbb{R}^{m_e}} \left\{ \frac{1}{2} \left\| \mathbf{\alpha}^e - \tilde{\mathbf{\alpha}}^e \right\|_2^2 + \frac{1}{\sigma} \lambda \mathbf{\alpha}^e (\max(\mathbf{\alpha}^e) - \min(\mathbf{\alpha}^e))^2 \right\}
\]
can be computed with \( O(m_e \log m_e) \) arithmetic operations.

We will now derive such an algorithm. To simplify the notation, we consider instead of \( \frac{1}{\sigma} F_e \) the function \( h : \mathbb{R}^m \to \mathbb{R} \) defined by
\[
h(\mathbf{\alpha}) = (\max(\mathbf{\alpha}) - \min(\mathbf{\alpha}))^2
\]
and show that \( \text{prox}_{\sigma h} (\mathbf{\alpha}) \), \( \mu > 0 \), can be computed with \( O(m \log m) \) arithmetic operations.

Let us fix \( \mathbf{\alpha} \in \mathbb{R}^m \). For every pair \( r, s \in [\min(\mathbf{\alpha}), \max(\mathbf{\alpha})] \) with \( r \geq s \), we define \( \alpha^{(r, s)} \) by
\[
\alpha^{(r, s)}_i = \begin{cases} r & \text{if } \alpha_i \geq r \\ \alpha_i & \text{if } \alpha_i \in (r, s) \\ s & \text{if } \alpha_i \leq s \end{cases}
\]
(12)

Clearly, if \( r = \max(\text{prox}_{\sigma h} (\mathbf{\alpha})) \) and \( s = \min(\text{prox}_{\sigma h} (\mathbf{\alpha})) \) then \( \alpha^{(r, s)} = \text{prox}_{\sigma h} (\mathbf{\alpha}) \). Hence, the above definition allows us to write the proximal problem in terms of the variables \( r, s \) since for
\[
(r, s) = \arg \min_{r, s} \left\{ \frac{1}{2} \left\| \alpha^{(r, s)} - \alpha^e \right\|_2^2 + \frac{\mu}{\sigma} (r - s)^2 \right\}
\]
(13)
we have
\[
\text{prox}_{\sigma h} (\mathbf{\alpha}) = \alpha^{(r, s)}.
\]

Our goal is now to find a minimizer of \((13)\). To this end, we first order \( \alpha \) in an increasing order which can be done in \( O(m \log m) \) arithmetic operations. W.l.o.g. we assume here that the components of \( \alpha \) are pairwise different. Moreover, we introduce the following notation. For \( r, s \in [\alpha_1, \alpha_m] \) there exist unique \( p, q \in \{1, \ldots, m\} \) characterized by \( \alpha_{m-p+1} = \min \{ \alpha_i : \alpha_i \geq r \} \) and \( \alpha_q = \max \{ \alpha_i : \alpha_i \leq s \} \). Thus, the directional partial derivatives w.r.t. \( r \) and \( s \) are given by
\[
\frac{\partial E_1}{\partial r} (r, s) = \sum_{i=m-p+1}^m (\alpha_i - r), \quad \frac{\partial E_1}{\partial s} (r, s) = \sum_{i=1}^q (s - \alpha_i).
\]
(14)

They tell us how much we increase \( E_1 \) by decreasing \( r \) and increasing \( s \), respectively. On the other hand both of these changes lead to a decrease in the energy \( E_2 \). More precisely, it holds that
\[
\frac{\partial E_2}{\partial r} (r, s) = \frac{\partial E_2}{\partial s} (r, s) = 2\mu (s - r).
\]
(15)

Thus, the main ideas behind our algorithm are as follows. Starting with \( r = \max(\alpha) \) and \( s = \min(\alpha) \), we decrease \( r \) and increase \( s \) keeping the two partial derivatives of \((14)\) equal. We stop when the sum of the partial derivatives vanishes. So, the optimal \( r, s \) are characterized by the system
\[
\sum_{i=m-p+1}^m (\alpha_i - r) = \sum_{i=1}^q (s - \alpha_i),
\]
(16)
\[
\sum_{i=m-p+1}^m (\alpha_i - r) + 2\mu (s - r) = 0.
\]
(17)
We will now generate a sequence of pairs \( r^{(k)}, s^{(k)} \) satisfying \( r^{(k)} \geq s^{(k)} \) and (16) for each \( k \). The corresponding indices needed to calculate the partial derivatives will be denoted by \( p^{(k)}, q^{(k)} \). The main procedure is described in the next lemma.

**Lemma 5.2.** Assume \( r^{(k)} \in (\alpha_{m-p^{(k)}}, \alpha_{m-p^{(k)}+1}] \) and \( s^{(k)} \in [\alpha_{q^{(k)}}, \alpha_{q^{(k)}+1}) \) and property (16) holds for \( (r^{(k)}, s^{(k)}) \). Then, we can either choose

\[
r^{(k+1)} = r^{(k)} - \frac{q^{(k)}}{p^{(k)}} (s^{(k+1)} - s^{(k)}) \quad \text{and} \quad s^{(k+1)} = \alpha_{q^{(k)}+1}
\]

(18)
or

\[
r^{(k+1)} = \alpha_{m-p^{(k)}} \quad \text{and} \quad s^{(k+1)} = s^{(k)} + \frac{p^{(k)}}{q^{(k)}} (r^{(k)} - r^{(k+1)})
\]

(19)

such that \( r^{(k+1)} \in [\alpha_{m-p^{(k)}}, \alpha_{m-p^{(k)}+1}] \), \( s^{(k+1)} \in (\alpha_{q^{(k)}}, \alpha_{q^{(k)}+1}) \) and (16) holds true for \( (r^{(k+1)}, s^{(k+1)}) \).

**Proof.** Property (16) for \( (r^{(k+1)}, s^{(k+1)}) \) means that

\[
\sum_{i=m-p^{(k)+1}}^{m} (\alpha_i - r^{(k+1)}) = \sum_{i=1}^{q^{(k)}} (s^{(k+1)} - \alpha_i).
\]

(20)

Since by assumption (16) holds for \( (r^{(k)}, s^{(k)}) \), equation (20) is equivalent to

\[
p^{(k)} (r^{(k+1)} - r^{(k)}) = q^{(k)} (s^{(k+1)} - s^{(k)}).
\]

If we set \((r^{(k+1)}, s^{(k+1)})\) according to (18) but \(r^{(k+1)} < \alpha_{m-p^{(k)}}\). Then we get

\[
r^{(k)} - \frac{q^{(k)}}{p^{(k)}} (\alpha_{q^{(k)}+1} - s^{(k)}) < \alpha_{m-p^{(k)}}
\]

\[
\Rightarrow s^{(k)} + \frac{p^{(k)}}{q^{(k)}} (r^{(k)} - \alpha_{m-p^{(k)})} < \alpha_{q^{(k)}+1},
\]

i.e., we can choose \((r^{(k+1)}, s^{(k+1)})\) according to (19) and vice versa. \(\square\)

After each computation of a new pair \((r^{(k+1)}, s^{(k+1)})\) we check if the left-hand side of (17) is smaller than zero (note that initially the left-hand side of (17) is negative and it is increasing for every iteration). If this is not the case, we found the intervals where the optimal values \(r\) and \(s\) lie in. Restricted to this domain the functional \(E_1 + E_2\) is a differentiable. Hence, we can compute \(r, s\) as follows.

**Lemma 5.3.** Assume that the optimal \(r, s\) of (13) fulfill \(r \in [\alpha_{m-p}, \alpha_{m-p+1}]\) and \(s \in [\alpha_q, \alpha_{q+1}]\). Then, it holds that

\[
s = \left(q + 2\mu - \sum_{i=m-p+1}^{m} \frac{2\mu^2}{\alpha_i + 2\mu}\right)^{-1} \left(\frac{2\mu}{p + 2\mu} \sum_{i=m-p+1}^{m} \alpha_i + \sum_{i=1}^{q} \alpha_i\right)
\]

\[
r = \frac{1}{2\mu} ((q + 2\mu)s - \sum_{i=1}^{q} \alpha_i).
\]

**Proof.** When restricted to \([\alpha_i, \alpha_{i+1}] \times [\alpha_j, \alpha_{j+1}]\), the function \((r, s) \mapsto E_i(r, s) + E_2(r, s)\) is a quadratic function in \((r, s)\). We can thus simply set the gradient to zero and solve the corresponding system of linear equations which yields the above result. \(\square\)

In conclusion, we obtain the following algorithm. Note that after the sorting, the algorithm takes in the order of \(m\) steps to compute the proximum which proves Proposition 5.1.

Hence, the corresponding PDHG algorithm can be formulated as follows.
Algorithm 4 – Solution of the proximal problem $\text{prox}_{\mu h}(\alpha)$

1. Sort $\alpha \in \mathbb{R}^m$ in increasing order.
2. Initialization: $r^{(0)} = \max(\alpha), s^{(0)} = \min(\alpha)$
3. while $\frac{2\mu}{\mu + |E|} (r^{(k)} - s^{(k)}) < 2\mu (r^{(k)} - s^{(k)})$ and $q^{(k)} + 1 \leq m - p^{(k)}$ do
4. Find $(r^{(k+1)}, s^{(k+1)})$ according to Lemma 5.2.
5. end while
6. Compute $r, s$ as described in Lemma 5.3.
7. Output: After restoring the original order, set

$$\left(\text{prox}_{\mu h}(\alpha)\right)_i = \begin{cases} r & \text{if } \alpha_i \geq r, \\ \alpha_i & \text{if } \alpha_i \in (r, s), \\ s & \text{if } \alpha_i \leq s, \end{cases} \text{ for } i = 1, \ldots, m.$$ 

Algorithm 5 PDHG for $\Omega_{H,2}$

1. Initialization: $f^{(0)} = \tilde{f}^{(0)} = 0, \theta \in [0, 1], \sigma, \tau > 0$ with $\sigma \tau < 1/\max_{i=1,\ldots,n} \{c_i\}$
2. repeat
3. $\alpha^{(k+1)} = \alpha^{(k)} + \sigma K e f^{(k)} - \text{prox}_{\epsilon \lambda E} (\alpha^{(k)} + \sigma K e \tilde{f}^{(k)}), \ e \in E$
4. $f^{(k+1)} = \text{prox}_{\epsilon \tau G} (f^{(k)} - \tau \sum_{e \in E} K^T e (\alpha^{(k+1)}))$
5. $\tilde{f}^{(k+1)} = f^{(k+1)} + \theta(f^{(k+1)} - f^{(k)})$
6. until relative duality gap $< \epsilon$
7. Output: $f^{(k+1)}$.

We solve the subproblems in line 3 via Algorithm 4. Note that the bound on the step sizes is now doubled, i.e., less restrictive since we have defined for each hyperedge one functional $F_e$ and not two as for $p = 1$, i.e.,

$$\|K\|^2 = \|K^T K\|^2 = \| \sum_{e \in E} K^T e K_e \|^2 = \max_{i=1,\ldots,n} \{c_i\}.$$ 

### 6 Experiments

As our literature review revealed, in practice the method of Zhou et al [11] is most often used for clustering and semi-supervised learning on hypergraphs. We compare to them on the selection of UCI datasets which are summarized in Table 2. Zoo, Mushrooms and 20Newsgroups3 have been used also in [11] and contain only categorical features. As in [11], a hyperedge of weight one is created by all data points which have the same value of a categorical feature. This is repeated for all possible values of every categorical feature. For covertype a subset of the features is numerical. These features are turned into categorical features by binning them into 10 bins of equal size. We create two datasets with each two classes (4,5) and (6,7) of the original dataset.

3This is a modified version by Sam Roweis of the original 20 newsgroups dataset available at http://www.cs.nyu.edu/~roweis/data/20news_w100.mat.

<table>
<thead>
<tr>
<th>Prop. \ Dataset</th>
<th>Zoo</th>
<th>Mushrooms</th>
<th>Covertype (4,5)</th>
<th>Covertype (6,7)</th>
<th>20Newsgroups</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of classes</td>
<td>7</td>
<td>2</td>
<td>2</td>
<td>4</td>
<td></td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>12240</td>
<td>37877</td>
<td>16242</td>
<td></td>
</tr>
<tr>
<td></td>
<td>101</td>
<td>8124</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>42</td>
<td>112</td>
<td>104</td>
<td>123</td>
<td>100</td>
</tr>
<tr>
<td></td>
<td>1717</td>
<td>170604</td>
<td>146880</td>
<td>454522</td>
<td>65451</td>
</tr>
<tr>
<td></td>
<td>10201</td>
<td>65999376</td>
<td>143008092</td>
<td>1348219153</td>
<td>53284642</td>
</tr>
</tbody>
</table>

Table 2: Datasets used for SSL and clustering. Note that the clique expansion leads for all datasets to a graph which is close to being fully connected as all datasets contain large hyperedges. For covertype (6,7) the weight matrix needs over 10GB of memory, the original hypergraph only 4MB.
Semi-supervised Learning. In [11], they suggest to use a regularizer induced by the normalized Laplacian $L_{CE}$ arising from the clique expansion

$$L_{CE} = I - D^{-\frac{1}{2}}_C W' H^3 D^{-\frac{1}{2}}_C,$$

where $D_C$ is a diagonal matrix with entries $d_C(i) = \sum_{e \in E} H_{i,e} w_i$ and $W' \in \mathbb{R}^{|E| \times |V|}$ is a diagonal matrix with entries $w'_i(e) = w_i / |e|$. The semi-supervised learning problem can then be formulated with $\lambda > 0$ as

$$\arg\min_{f \in \mathcal{F}} \left\{ \| f - Y \|_2^2 + \lambda \langle f, L_{CE} f \rangle \right\}.$$ 

The advantage of this formulation is that the solution can be found via a linear system. However, as Table 2 indicates the obvious downside is that $L_{CE}$ is a potentially very dense matrix and thus one needs in the worst case $|V|^2$ memory and $O(|V|^3)$ computations. This is in contrast to our method which needs $2 \sum_{e \in E} |e| + |V|$ memory. For the largest example (covertype 6,7), where the clique expansion fails due to memory problems, our method takes 30-100s (depending on $\lambda$). We stop our method for all experiments when we achieve a relative duality gap of $10^{-6}$.

In the experiments we do 10 trials for different numbers of labeled points. We choose the regularization parameter for both methods from the set $10^{-k}$, where $k = \{0,1,2,3,4,5\}$ via 5-fold cross validation. The resulting errors and standard deviations can be found in the following table (first row lists the no. of labeled points).

<table>
<thead>
<tr>
<th>Zoo</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>160</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al</td>
<td>35.1±2.7</td>
<td>30.4±7.9</td>
<td>40.7±14.2</td>
<td>29.7±8.8</td>
<td>32.9±16.8</td>
<td>27.6±10.8</td>
<td>25.3±14.4</td>
<td></td>
</tr>
<tr>
<td>$\Omega_{L,1}$</td>
<td>2.9±3.0</td>
<td>1.4±2.2</td>
<td>2.3±4.8</td>
<td>2.9±3.8</td>
<td>0.9±1.4</td>
<td>0.8±1.7</td>
<td>1.2±1.8</td>
<td>1.6±2.9</td>
</tr>
<tr>
<td>$\Omega_{L,2}$</td>
<td>2.3±4.9</td>
<td>1.5±4.4</td>
<td>2.9±4.5</td>
<td>3.9±5.5</td>
<td>6.4±2.7</td>
<td>6.3±2.5</td>
<td>4.5±1.8</td>
<td>4.4±2.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Music</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>160</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al</td>
<td>16.5±12.8</td>
<td>10.3±4.4</td>
<td>9.5±2.7</td>
<td>10.3±2.0</td>
<td>9.0±4.8</td>
<td>8.8±4.4</td>
<td>8.8±4.4</td>
<td>9.3±1.8</td>
</tr>
<tr>
<td>$\Omega_{H,1}$</td>
<td>19.5±10.5</td>
<td>10.8±3.7</td>
<td>7.4±3.8</td>
<td>5.6±1.9</td>
<td>5.7±2.2</td>
<td>5.4±2.4</td>
<td>4.9±3.8</td>
<td>5.6±3.8</td>
</tr>
<tr>
<td>$\Omega_{H,2}$</td>
<td>18.4±7.4</td>
<td>9.8±4.5</td>
<td>9.5±5.5</td>
<td>6.4±2.7</td>
<td>6.3±2.5</td>
<td>4.5±1.8</td>
<td>4.4±2.1</td>
<td>3.0±0.6</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>covertype</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>160</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al</td>
<td>18.9±4.6</td>
<td>18.3±5.2</td>
<td>17.2±0.7</td>
<td>16.6±4.4</td>
<td>15.6±2.2</td>
<td>18.4±6.1</td>
<td>19.2±4.0</td>
<td>20.4±2.9</td>
</tr>
<tr>
<td>$\Omega_{H,1}$</td>
<td>21.4±0.9</td>
<td>17.6±2.7</td>
<td>12.6±4.3</td>
<td>7.6±3.5</td>
<td>8.3±3.7</td>
<td>4.5±3.6</td>
<td>2.6±1.6</td>
<td>1.5±1.3</td>
</tr>
<tr>
<td>$\Omega_{H,2}$</td>
<td>20.7±2.0</td>
<td>16.1±4.1</td>
<td>10.9±4.9</td>
<td>5.9±3.7</td>
<td>5.6±3.4</td>
<td>3.8±3.1</td>
<td>2.2±1.8</td>
<td>1.0±1.1</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>covertype</th>
<th>20</th>
<th>40</th>
<th>60</th>
<th>80</th>
<th>100</th>
<th>120</th>
<th>160</th>
<th>200</th>
</tr>
</thead>
<tbody>
<tr>
<td>Zhou et al</td>
<td>40.6±8.9</td>
<td>6.4±10.4</td>
<td>3.6±4.4</td>
<td>3.4±2.5</td>
<td>1.8±0.8</td>
<td>1.3±0.9</td>
<td>0.9±0.4</td>
<td>1.2±0.8</td>
</tr>
<tr>
<td>$\Omega_{H,1}$</td>
<td>25.2±18.3</td>
<td>4.3±9.0</td>
<td>2.1±2.0</td>
<td>2.2±1.4</td>
<td>1.4±1.1</td>
<td>1.0±0.8</td>
<td>0.7±0.4</td>
<td>1.1±0.8</td>
</tr>
<tr>
<td>$\Omega_{H,2}$</td>
<td>25.2±18.3</td>
<td>4.3±9.0</td>
<td>2.1±2.0</td>
<td>2.2±1.4</td>
<td>1.4±1.1</td>
<td>1.0±0.8</td>
<td>0.7±0.4</td>
<td>1.1±0.8</td>
</tr>
</tbody>
</table>

We observe that our SSL methods based on $\Omega_{H,p}, p = 1, 2$ outperform consistently the clique expansion technique of Zhou et al [11] on all datasets except 20newsgroups4. However, 20newsgroups is a very difficult dataset as only 10,267 out of the 16,242 data points are different which leads to a minimum possible error of 9.6%. A method based on pairwise interaction such as the clique expansion can better deal with such label noise as the hyperedges for this dataset are very large and thus accumulate the label noise. On all other datasets we observe that incorporating hypergraph structure leads to much better results. As expected our squared TV functional ($p = 2$) outperforms slightly the total variation on hypergraph ($p = 1$) even though the difference is small. However, as $\Omega_{H,2}$ reduces to the standard regularization based on the graph Laplacian, which is known to work well, we recommend this for general SSL on hypergraphs.

Clustering. We use the normalized hypergraph cut as clustering objective. For more than two clusters we recursively partition the hypergraph until the desired number of clusters is reached. For comparison we use the normalized spectral clustering approach based on the Laplacian $L_{CE}$ [11](clique expansion). The first part (first 6 columns) of the following table shows the clustering errors of both methods as well as the normalized cuts achieved by these methods on the hypergraph and on the graph resulting from the clique expansion. Moreover, we show results (last 4 columns) which are obtained based on a kNN graph (unit weights) which is built based on the Hamming distance (note that we have categorical features) in order to check if the hypergraph modeling of the

\footnote{Note that the results on 20newsgroups differ from [11] and communication with the authors of [11] could not clarify the difference.}
problem is actually useful compared to a standard similarity based graph construction. The number $k$ is chosen as the smallest number for which the graph becomes connected and we compare results of normalized $1$-spectral clustering [14] and the standard spectral clustering [22]. Note that the employed hypergraph construction has no free parameter.

First, we observe that optimizing the normalized cuts on hypergraphs yields better or similar results in terms of clustering errors compared to the clique expansion (except for 20-newsgroup for the same reason given in the previous paragraph). The improvement is significant in case of Mushrooms while for Zoo our clustering error is slightly higher. However, we always achieve smaller normalized hypergraph cuts. Moreover, our method sometimes has even smaller cuts on the graphs resulting from the clique expansion, although it does not directly optimize this objective in contrast to [11]. Again, we could not run the method of [11] on covertype (6,7) since the weight matrix is very dense. Second, the comparison to a standard graph-based approach where the similarity structure is obtained using the Hamming distance on the categorical feature shows that using hypergraph structure is indeed useful. Nevertheless, we think that there is room for improvement regarding the construction of the hypergraph.

Acknowledgments

M.H. would like to acknowledge support by the ERC Starting Grant NOLEPRO and L.J. acknowledges support by the DFG SPP-1324.

References


<table>
<thead>
<tr>
<th>Dataset</th>
<th>Clustering Error %</th>
<th>Hypergraph Ncut</th>
<th>Graph(C.E) Ncut</th>
<th>Clustering Error %</th>
<th>ANN-Graph Ncut</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mushrooms</td>
<td>10.98 15.84</td>
<td>0.0301 0.0301</td>
<td>0.0301 0.0301</td>
<td>4.82 4.82</td>
<td>1e-4 1e-4</td>
</tr>
<tr>
<td>Zoo</td>
<td>15.84 0.0011</td>
<td>0.0013 0.0013</td>
<td>0.0013 0.0013</td>
<td>5.94 5.94</td>
<td>1e-4 1e-4</td>
</tr>
<tr>
<td>20-newsgroup</td>
<td>15.84 0.0011</td>
<td>0.0013 0.0013</td>
<td>0.0013 0.0013</td>
<td>5.94 5.94</td>
<td>1e-4 1e-4</td>
</tr>
<tr>
<td>covertype (4,5)</td>
<td>15.84 0.0011</td>
<td>0.0013 0.0013</td>
<td>0.0013 0.0013</td>
<td>5.94 5.94</td>
<td>1e-4 1e-4</td>
</tr>
<tr>
<td>covertype (6,7)</td>
<td>15.84 0.0011</td>
<td>0.0013 0.0013</td>
<td>0.0013 0.0013</td>
<td>5.94 5.94</td>
<td>1e-4 1e-4</td>
</tr>
</tbody>
</table>
Reports


