

# Dynamic Hypergraph Structure Learning

Zizhao Zhang, Haojie Lin, Yue Gao\*

BNRist, KLISS, School of Software, Tsinghua University, China.

zz-zh14@mails.tsinghua.edu.cn, haojie.lin@outlook.com, gaoyue@tsinghua.edu.cn

## Abstract

In recent years, hypergraph modeling has shown its superiority on correlation formulation among samples and has wide applications in classification, retrieval, and other tasks. In all these works, the performance of hypergraph learning highly depends on the generated hypergraph structure. A good hypergraph structure can represent the data correlation better, and vice versa. Although hypergraph learning has attracted much attention recently, most of existing works still rely on a static hypergraph structure, and little effort concentrates on optimizing the hypergraph structure during the learning process. To tackle this problem, we propose the first dynamic hypergraph structure learning method in this paper. In this method, given the originally generated hypergraph structure, the objective of our work is to simultaneously optimize the label projection matrix (the common task in hypergraph learning) and the hypergraph structure itself. On one hand, the label projection matrix is related to the hypergraph structure in our formulation, similar to traditional methods. On the other hand, different from the existing hypergraph generation methods based on the feature space only, the hypergraph structure optimization in our formulation utilizes the data correlation from both the label space and the feature space. Then, we alternatively learn the optimal label projection matrix and the hypergraph structure, leading to a dynamic hypergraph structure during the learning process. We have applied the proposed method in the tasks of 3D shape recognition and gesture recognition. Experimental results on four public datasets show better performance compared with the state-of-the-art methods. We note that the proposed method can be further applied in other tasks.

## 1 Introduction

Hypergraph is composed of a vertex set and a hyperedge set. In a hypergraph, each hyperedge can connect any number of vertices, and the hyperedge can be easily expanded, which leads to a flexible edge degree for hypergraph. Therefore, the hypergraph model is able to formulate the high-order correlation of data compared with simple graph and other linear comparison methods.

Due to this advantage, hypergraph learning method [Zhou and Schölkopf, 2007] has attracted much attention in recent years and has been applied in many tasks [Zhang *et al.*, 2017; Zhu *et al.*, 2017b], such as image retrieval [Huang *et al.*, 2010; Zhu *et al.*, 2017a], social network analysis [Zhao *et al.*, 2018], object classification [Gao *et al.*, 2012], image segmentation [Huang *et al.*, 2009], hyperspectral image analysis [Luo *et al.*, 2018], person re-identification [An *et al.*, 2017], multimedia recommendation [Zhu *et al.*, 2015] and visual tracking [Du *et al.*, 2017]. In these applications, usually the relationship of the data for processing is formulated in a hypergraph structure, where each vertex denotes the data and the hyperedges represent some kind of connections. Then, the hypergraph learning is conducted as a label propagation process on the hypergraph to obtain the label projection matrix [Liu *et al.*, 2017a] or as a spectral clustering [Li and Milenkovic, 2017] in different tasks.

In these methods, the quality of the hypergraph structure plays an important role for data modelling. A well constructed hypergraph structure can represent the data correlation accurately, yet leading to better performance, while an inappropriate hypergraph structure could make the situation worse. To generate a better hypergraph structure, many approaches have been proposed in recent years, such as  $k$ -nn method [Huang *et al.*, 2009], clustering-based method [Gao *et al.*, 2012], spare representation method [Liu *et al.*, 2017b], etc. In existing methods, after the hypergraph has been constructed, it never changes during the learning process, leading to a static hypergraph structure learning mechanism. However, it is uneasy to guarantee that the generated hypergraph structure is optimal and suitable for all applications.

In recent years, there have been some attempts to modify the hypergraph components, such as learning the hyperedge weights [Gao *et al.*, 2012] and the sub-hypergraph

---

\*indicates corresponding author

weights [Gao *et al.*, 2013]. However, these methods only make slightly maintenance on the original hypergraph structure, and cannot repair the connections which are inappropriate or even contain errors. Confronting this challenge, it is urgent to investigate the hypergraph structure optimization during the learning process, leading to a dynamic hypergraph structure learning scheme.

In this paper, we propose a dynamic hypergraph structure learning method, which can jointly learn the hypergraph structure and label projection matrix simultaneously. In a hypergraph, its structure is represented by the incidence matrix, which records the connections and the connection strength between vertices and hyperedges. The objective here is to optimize the incidence matrix during the learning process, yet leading to a better hypergraph structure accordingly. Given the data, an initial hypergraph structure can be generated by selecting a suitable hypergraph construction method for the application. In our formulation, we propose to optimize the hypergraph structure from two spaces, *i.e.*, the data label space and the data feature space. With the initial hypergraph, a transductive learning process can be conducted to propagate the vertex labels on the hypergraph. In each round, after the optimization of the label projection matrix, we further update the incidence matrix considering the data correlation on both the label space and the feature space. This process repeats until both the hypergraph structure and the label projection matrix are stable. During this process, the hypergraph structure can be dynamically updated, leading to a dynamic hypergraph strategy. In this way, we can achieve both optimal label projection matrix (the objective for the application) and updated hypergraph structure (the data correlation modelling) under a dual optimization framework.

We have applied our dynamic hypergraph structure learning method on the tasks of 3D shape recognition and gesture recognition. For object classification, experiments have been conducted on two public benchmarks, *i.e.*, the National Taiwan University (NTU) 3D shape dataset [Chen *et al.*, 2003] and the Engineering Shape Benchmark (ESB) [Jayanti *et al.*, 2006]. For gesture recognition, experiments have been conducted on the public MSR Gesture 3D dataset (MSRGesture3D) and a motion gesture dataset collected by Huazhong University of Science and Technology (Gesture3DMotion). Experimental results demonstrate better performance of the proposed method compared with traditional hypergraph learning methods and other state-of-the-art methods, indicating the better data correlation modeling ability.

The main contributions of our work are two-fold:

1. We propose the first dynamic hypergraph structure learning method. To the best of our knowledge, this is the first attempt to jointly conduct hypergraph structure updating and hypergraph learning, which is accomplished by a dual optimization process. For hypergraph structure optimization, we propose to update the hypergraph structure from both the label space and the feature space, different from traditional methods which are mainly based on the feature space.
2. We have applied the proposed method on 3D shape recognition and gesture recognition and conducted ex-

tensive evaluations to demonstrate the effectiveness of the proposed method. We have observed that traditional hypergraph hyperedge weight updating method has the limitation on improving the hypergraph learning performance, and the dynamic updating of hypergraph structure has shown consistent performance improvement.

The rest of this paper is organized as follows. Section 2 introduces the related work on hypergraph learning. Section 3 presents the proposed dynamic hypergraph structure learning method. The applications and experimental results are provided in Section 4. We conclude this paper in Section 5.

## 2 Related Work

In this section, we briefly review existing works on hypergraph learning and its applications, such as video object segmentation, scene recognition, 3D object retrieval and classification.

In hypergraph learning, hypergraph construction is the first step for data modeling. Existing works can be divided into feature-based methods and representation-based methods. In feature-based methods, the hyperedge targets on exploring the nearest neighbors in the feature space, based on  $k$ -nn or a search radius. The  $k$ -nn scheme [Huang *et al.*, 2009] selects a set number of nearest neighbors for each vertex to generate a hyperedge. The search radius scheme, while, sets a pre-defined search radius and all vertices in the radius are connected by one hyperedge. However, it is challenging to determine an optimal number of nearest neighbors or the search radius, which may be sensitive to noise and limit the performance of data modeling. Different from the feature-based methods, representation-based method [Liu *et al.*, 2017b] aims to exploit the relation among vertices through feature reconstruction. Given a centroid vertex, the sparse representation was conducted to represent the centroid vertex by its neighbors, and then only the vertices with non-zero coefficients to the centroid vertex were used to construct the hyperedge. To deal with multi-modal data, multi-hypergraph structure [Gao *et al.*, 2012; Zhu *et al.*, 2015] has also been introduced corresponding to different modalities or features.

In [Zhou and Schölkopf, 2007], hypergraph learning was introduced as a propagation process through the hypergraph structure. The transductive inference on hypergraph aims to minimize the label differences between vertices which are connected by more and stronger hyperedges. To further improve the hypergraph structure, recent attention has been spent on learning the weights of hyperedges, where the hyperedges could have different importance on data modeling. To learn optimal hyperedge weights, a  $\ell_2$  regularizer on the weights was introduced in [Gao *et al.*, 2013]. Hwang *et al.* [Hwang *et al.*, 2008] further considered the correlation among hyperedges by assuming that highly correlated hyperedges should share similar weights. Regarding multi-hypergraph, a sub-hypergraph weight learning method was introduced in [Gao *et al.*, 2012] to assign weights for different sub-hypergraphs. We note that although these works can update the hyperedge weights or the sub-hypergraph weights, they cannot repair the connections which are inappropriate or even contain errors because the hypergraph structure, *i.e.*, the

incidence matrix is static in all these works.

Hypergraph learning has been widely applied in many computer vision tasks. In [Huang *et al.*, 2010], the hypergraph structure was employed to formulate the relationship among images based on visual features for image retrieval. The hypergraph structure was used for video object segmentation [Huang *et al.*, 2009], where each vertex denoted over-segmented image patched and the spatial-temporal neighborhood correlation was used for hypergraph construction. Hypergraph learning has also been used for image classification and ranking [Gao *et al.*, 2013; Zhu *et al.*, 2015], and 3D object classification [Gao *et al.*, 2012].

We can notice that existing hypergraph learning methods are based on a fixed (static) hypergraph structure. These works either focus on the learning of label projection matrix or the learning of hyperedge/hypergraph weights. Given an inaccurate initial hypergraph, it is impossible for existing methods to obtain an optimal data modelling. Therefore, static hypergraph learning still has the limitation on complex data correlation modelling.

### 3 Dynamic Hypergraph Structure Learning

In this section, we detailed introduce the proposed dynamic hypergraph structure learning algorithm. Figure 1 illustrates the general framework of our proposed method. Given the data with corresponding features, our objective is to learn an optimal hypergraph structure, which could better explore the high-order relationship among the data, and also the label project matrix, which is used for discriminating the testings to different categories. Firstly, a hypergraph can be generated to formulate the data correlation. We note that any hypergraph construction method can be used in our framework, and we take the  $k$ -nn scheme as an example in this paper. Given the initial hypergraph, we then jointly learn the label projection matrix and the hypergraph structure under a dual optimization framework. In each round, a transductive learning process can be conducted to propagate the vertex labels on the hypergraph. After the label projection matrix has been obtained, we further learn the incidence matrix considering the data correlation on both the label space and the feature space. This process repeats until the objective function doesn't descend. During this process, the hypergraph structure can be dynamically updated. We then have both optimal label projection matrix and updated hypergraph structure. The former one can be used for data classification, retrieval and other applications.

#### 3.1 Hypergraph Construction

For any application using hypergraph learning methods, the first step is to construct the corresponding hypergraph structure. Given the data, including  $n$  labeled training samples  $\mathcal{Q} = \{\mathcal{S}_1, \mathcal{S}_2, \dots, \mathcal{S}_n\}$  and  $m$  unlabeled testing samples  $\mathcal{D} = \{\mathcal{S}_{n+1}, \mathcal{S}_{n+2}, \dots, \mathcal{S}_{n+m}\}$ , here let  $x_i$  denote the feature vector of the  $i$ -th sample and  $\mathbf{X} = \{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_{n+m}\}$ . A hypergraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$  can be constructed to formulate the relationship among these samples, where  $\mathcal{V}$  is the set of vertices,  $\mathcal{E}$  is the set of hyperedges and  $\mathbf{W}$  is a diagonal matrix where each diagonal element denotes the hyperedge

weights. In  $\mathcal{V}$ , each vertex denotes one sample in  $\{\mathcal{Q}, \mathcal{D}\}$ , and there are  $n + m$  vertices in total. In this paper, we take the  $k$ -nn hyperedge generation method as the example. We note that other hypergraph generation methods can be also used in our framework. In  $\mathcal{E}$ , the hyperedges are generated following a  $k$ -nn approach. Each time, one vertex is selected as the centroid and a corresponding hyperedge  $e$  connects the centroid and its  $k$  nearest neighbor vertices. The incidence matrix  $\mathbf{H}$  of hypergraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$  represents the connections between the vertices and the hyperedges, which can be generated as

$$\mathbf{H}(v, e) = \begin{cases} \exp\left(-\frac{d(v, v_c)^2}{\bar{d}^2}\right) & \text{if } v \in e \\ 0 & \text{if } v \notin e \end{cases}, \quad (1)$$

where  $d(v, v_c)$  is the distance between a vertex  $v$  and the centroid vertex  $v_c$  in hyperedge  $e$ . and  $\bar{d}$  is the average distance between subjects. Accordingly, two diagonal matrices  $\mathbf{D}_v$  and  $\mathbf{D}_e$  can be generated with every entry along the diagonal corresponds to the vertex degree and hyperedge degree, respectively. They can be written in the matrix form as  $\mathbf{D}_v = \text{diag}(\mathbf{H}\mathbf{W}\mathbf{1})$  and  $\mathbf{D}_e = \text{diag}(\mathbf{1}^T\mathbf{H})$ , respectively. All the hyperedges are given equal weights here.

#### 3.2 The Formulation of Dynamic Hypergraph Structure Learning

In a general setting of hypergraph learning, the label projection matrix is the learning target given the initial labels of the hypergraph structure.

Here we let  $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \dots, \mathbf{y}_{n+m}\}$  and  $\mathbf{y}_i$  denotes the initial label vector of the  $i$ -th sample (vertex). For labeled samples, the  $j$ -th element is 1 if the  $i$ -th sample belongs to the  $j$ -th class, and it is 0 otherwise. For unlabeled samples, all elements are set to 0.5, indicating that we have no prior information about the category of these samples. Let  $\mathbf{f}_i$  denotes the to-be-learned label projection vector of the  $i$ -th sample and  $\mathbf{F} = \{\mathbf{f}_1, \mathbf{f}_2, \dots, \mathbf{f}_{n+m}\}$  is the label project matrix.

The joint learning of the label projection matrix  $\mathbf{F}$  and the hypergraph structure, *i.e.*, the incidence matrix  $\mathbf{H}$ , should satisfy the following three conditions.

First, for the label projection matrix  $\mathbf{F}$ , it should be smooth on the hypergraph structure  $\mathbf{H}$ . The more and stronger connections between two vertices, the higher probabilities that they share similar label projection vector. This is a commonly used hypergraph regularizer and it is related to both  $\mathbf{F}$  and  $\mathbf{H}$ . It can be written as

$$\begin{aligned} & \Psi(\mathbf{F}, \mathbf{H}) \\ &= \frac{1}{2} \sum_{c=1}^{n_c} \sum_{e \in \mathcal{E}} \sum_{u, v \in \mathcal{V}} \frac{\mathbf{W}(e)\mathbf{H}(u, e)\mathbf{H}(v, e)}{\delta(e)} \left( \frac{\mathbf{F}(u, c)}{\sqrt{d(u)}} - \frac{\mathbf{F}(v, c)}{\sqrt{d(v)}} \right)^2 \\ &= \sum_{c=1}^{n_c} \sum_{e \in \mathcal{E}} \sum_{u, v \in \mathcal{V}} \frac{\mathbf{W}(e)\mathbf{H}(u, e)\mathbf{H}(v, e)}{\delta(e)} \left( \frac{\mathbf{F}(u, c)^2}{d(u)} - \frac{\mathbf{F}(u, c)\mathbf{F}(v, c)}{\sqrt{d(u)d(v)}} \right) \\ &= \sum_{c=1}^{n_c} \sum_{u \in \mathcal{V}} \mathbf{F}(u, c)^2 \sum_{e \in \mathcal{E}} \frac{\mathbf{W}(e)\mathbf{H}(u, e)}{d(u)} \sum_{v \in \mathcal{V}} \frac{\mathbf{H}(v, e)}{\delta(e)} \\ &\quad - \sum_{e \in \mathcal{E}} \sum_{u, v \in \mathcal{V}} \frac{\mathbf{F}(u, c)\mathbf{H}(u, e)\mathbf{W}(e)\mathbf{H}(v, e)\mathbf{F}(v, c)}{\sqrt{d(u)d(v)\delta(e)}} \\ &= \text{tr} \left( \left( \mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \right) \mathbf{F} \mathbf{F}^T \right), \end{aligned} \quad (2)$$

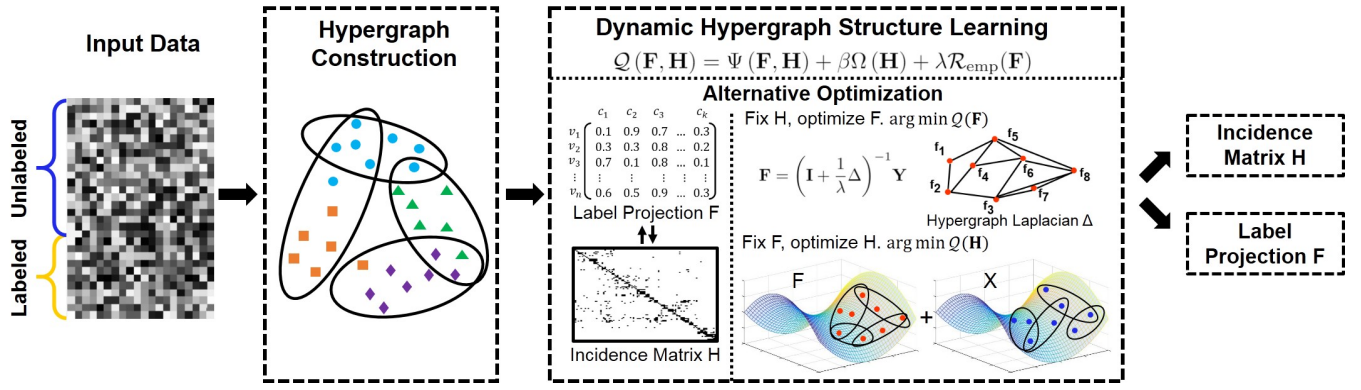


Figure 1: The framework of our proposed dynamic hypergraph structure learning method.

where  $n_c$  is the number of categories.

Second, for the hypergraph structure  $\mathbf{H}$ , it should also be smooth on the data from both the label space and the feature space. In the label space, the regularizer on  $\mathbf{H}$  is the same as Eq. (2). In the feature space, the hypergraph structure  $\mathbf{H}$  should also preserve the smoothness for the input features  $\mathbf{X}$ . Similar to Eq. (2), the constraint of  $\mathbf{H}$  on  $\mathbf{X}$  can be written as

$$\Omega(\mathbf{H}) = \text{tr}\left(\left(\mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\right)\mathbf{X}\mathbf{X}^T\right). \quad (3)$$

The third condition is the empirical loss, which comes from  $\mathbf{F}$  and  $\mathbf{Y}$  and can be written as

$$\mathcal{R}_{\text{emp}}(\mathbf{F}) = \|\mathbf{F} - \mathbf{Y}\|_F^2. \quad (4)$$

By summarizing the above optimization requirement, the objective function for dynamic hypergraph structure learning can be written as the following dual optimization problem:

$$\begin{aligned} \arg \min_{\mathbf{F}, \mathbf{0} \leq \mathbf{H} \leq \mathbf{1}} \mathcal{Q}(\mathbf{F}, \mathbf{H}) &= \Psi(\mathbf{F}, \mathbf{H}) + \beta\Omega(\mathbf{H}) + \lambda\mathcal{R}_{\text{emp}}(\mathbf{F}) \\ &= \text{tr}\left(\left(\mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\right)(\mathbf{F}\mathbf{F}^T + \beta\mathbf{X}\mathbf{X}^T)\right) \\ &\quad + \lambda\|\mathbf{F} - \mathbf{Y}\|_F^2, \end{aligned} \quad (5)$$

where  $\beta$  and  $\lambda$  are parameters to balance different components in the objective function.

### 3.3 Solution

Here, we optimize the objective function in Eq.(5) to learn the label projection matrix  $\mathbf{F}$  and the hypergraph structure  $\mathbf{H}$  simultaneously. We employ the alternative optimization scheme to update each variable when fixing the other until the objective function doesn't descend.

#### Optimization on $\mathbf{F}$

We first fix the incidence matrix  $\mathbf{H}$ , and the sub-problem on optimizing  $\mathbf{F}$  can be written as

$$\begin{aligned} \arg \min_{\mathbf{F}} \mathcal{Q}(\mathbf{F}) &= \Psi(\mathbf{F}) + \lambda\mathcal{R}_{\text{emp}}(\mathbf{F}) \\ &= \text{tr}(\Delta\mathbf{F}\mathbf{F}^T) + \lambda\|\mathbf{F} - \mathbf{Y}\|^2, \end{aligned} \quad (6)$$

where  $\Delta = \mathbf{I} - \mathbf{D}_e^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}$  can be regarded as the normalized hypergraph Laplacian in a matrix form. According to [Zhou and Schölkopf, 2007],  $\mathbf{F}$  can be obtained directly as

$$\mathbf{F} = \left(\mathbf{I} + \frac{1}{\lambda}\Delta\right)^{-1} \mathbf{Y}. \quad (7)$$

#### Optimization on $\mathbf{H}$

Then, we fix  $\mathbf{F}$  and optimize  $\mathbf{H}$ , and the learning objective function can be written as

$$\begin{aligned} \arg \min_{\mathbf{0} \leq \mathbf{H} \leq \mathbf{1}} \mathcal{Q}(\mathbf{H}) &= \Psi(\mathbf{H}) + \beta\Omega(\mathbf{H}) \\ &= \text{tr}\left(\left(\mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\right)\mathbf{K}\right) \end{aligned} \quad (8)$$

where  $\mathbf{K} = \mathbf{F}\mathbf{F}^T + \beta\mathbf{X}\mathbf{X}^T$ . Note that Eq. (8) is a bound constraint optimization problem and it can be solved by using the projected gradient method. The derivation with respect to  $\mathbf{H}$  can be written as:

$$\begin{aligned} \nabla \mathcal{Q}(\mathbf{H}) &= \mathbf{J} \left( \mathbf{I} \otimes \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{K} \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \right) \mathbf{W} \mathbf{D}_e^{-2} \\ &\quad + \mathbf{D}_v^{-\frac{3}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{K} \mathbf{J} \mathbf{W} \\ &\quad - 2 \mathbf{D}_v^{-\frac{1}{2}} \mathbf{K} \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1}, \end{aligned} \quad (9)$$

where  $\mathbf{J} = \mathbf{1}\mathbf{1}^T$ . The detailed derivation is included in Appendix A for reference. Then,  $\mathbf{H}$  can be updated by iterating

$$\mathbf{H}_{k+1} = \mathbf{P}[\mathbf{H}_k - \alpha \nabla \mathcal{Q}(\mathbf{H}_k)], \quad (10)$$

where  $\alpha$  is the optimal step size at each iteration and  $\mathbf{P}$  is the projection onto the feasible set  $\{\mathbf{H} | 0 \leq \mathbf{H} \leq \mathbf{1}\}$ . Here we set  $\mathbf{P}$  as

$$\mathbf{P}[h_{ij}] = \begin{cases} h_{ij} & \text{if } 0 \leq h_{ij} \leq 1 \\ 0 & \text{if } h_{ij} < 0 \\ 1 & \text{if } h_{ij} > 1 \end{cases}. \quad (11)$$

We note that other selection of  $\mathbf{P}$  can be also used in our framework.

With the updated  $\mathbf{H}$ , we can further optimize the label projection matrix  $\mathbf{F}$ . In this way, we can alternatively optimize  $\mathbf{F}$  and  $\mathbf{H}$  until the objective function doesn't descend. The overall workflow of the proposed dynamic hypergraph structure learning method is shown in Algorithm 1.

**Algorithm 1** Dynamic Hypergraph Structure Learning

**Input:** training data set  $\mathcal{Q}$ , testing data set  $\mathcal{D}$ , maximal iteration  $k$ , and parameters  $\beta$  and  $\lambda$

**Output:** label projection matrix  $\mathbf{F}$  and hypergraph structure  $\mathbf{H}$

- 1: Construct the initial hypergraph  $\mathcal{G} = (\mathcal{V}, \mathcal{E}, \mathbf{W})$ .
- 2: Initialize the learning rate  $\alpha$ .
- 3: **for**  $i = 0 \rightarrow k - 1$  **do**
- 4:     Fix  $\mathbf{H}$ , and update  $\mathbf{F}$  by Eq. (6).
- 5:     Fix  $\mathbf{F}$ , and update  $\mathbf{H}$  by Eq. (8).
- 6: **end for**
- 7: **return**  $\mathbf{F}, \mathbf{H}$

**4 Applications and Experiments**

In this section, we apply our proposed dynamic hypergraph structure learning method in two tasks of 3D object classification and gesture recognition. We conduct experiments and comparisons on two 3D object datasets and two gesture datasets to evaluate the performance, respectively.

**4.1 3D Shape Recognition**

In 3D shape recognition, given a set of 3D shapes for the training and the testing data, the shape features are first extracted. A hypergraph structure can be constructed and the dual optimization procedure can be conducted for shape recognition. The shape will be categorized to the category based on the label projection matrix.

To evaluate the performance of the proposed method on 3D shape recognition, we have conducted experiments on the National Taiwan University 3D shape dataset (NTU) [Chen *et al.*, 2003] and the Engineering Shape Benchmark (ESB) [Jayanti *et al.*, 2006]. In the NTU dataset, 2020 3D shapes from 67 categories are selected, such as *bomb, bottle, car, chair, cup, door, map, plane, sword, table, tank, and truck*. In the ESB dataset, 866 shapes from 43 categories are included, such as *back doors, bracket like parts, clips, contact switches, curved housings, thin plates, and bearing blocks*.

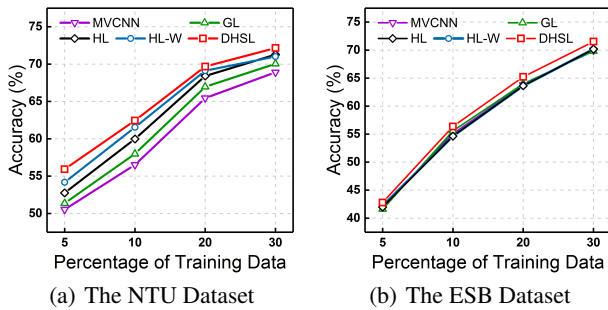


Figure 2: Experimental comparison of different methods on view-based object classification.

In our experiments, a set of data are selected for training and all other data are used for testing. The training data varies from 5% to 30% of all the data for each category respectively (note that for each category, at least one sample is selected for training). This training/testing data selection process repeats

10 times and the average recognition performance is reported. In these experiments, we empirically set the parameter  $\beta$  and  $\lambda$  as 10 and 1 on both datasets, respectively.

In experiments, the recent 3D shape recognition method, *i.e.*, multi-view convolutional neural networks (MVCNN) [Su *et al.*, 2015] is used for object feature extraction method and also for comparison. In MVCNN, 12 different views of each 3D shape are used for feature extraction, and it has shown the state-of-the-art performance of 3D shape recognition. In this task, we also compare our proposed dynamic hypergraph structure learning method (DHSL) with the graph-based learning (GL) [Zhou *et al.*, 2003], the traditional hypergraph learning (HL) [Zhou and Schölkopf, 2007] and the hypergraph learning with hyperedge weight learning (HL-W) [Gao *et al.*, 2013]. In GL, the normalized graph laplacian is explored. The radius parameter in regularization is tuned to the optimal value. In HL, the hypergraph is constructed in the same way as the proposed method, and traditional hypergraph learning is conducted for shape recognition. In HL-W, the hyperedge weights are also learned during the learning process. Note that GL, HL, HL-W and DHSL also use the MVCNN features as the shape features for hypergraph construction.

Experimental results on the two datasets are demonstrated in Figure 2. From these results, we can have the following observations:

1. Compared with the state-of-the-art method MVCNN, DHSL can achieve better performance on both datasets. For example, when 5%, 10%, 20% and 30% data are used for training, DHSL achieves gains of 10.69%, 10.5%, 6.45%, and 4.73% compared with MVCNN on the NTU dataset.
2. Compared with traditional learning methods, DHSL achieves better performance on both datasets. For example, compared with GL, HL and HL-W, DHSL achieve gains of 7.71%, 4.12% and 2.1% with 10% training data on the NTU dataset.

**4.2 Gesture Recognition**

In the task of gesture recognition, a gesture is represented by a depth sequence captured by depth sensors. Gesture recognition refers to classify the depth sequences into existing categories. Given a set of depth sequences, we first extract gesture features, and we then construct a hypergraph to model the high-order relationship among the data. After that, our method can conduct learning for gesture recognition.

To validate the proposed DHSL method on gesture recognition, we have conducted experiments on the MSR gesture 3D dataset (MSRGesture3D) [Wang *et al.*, 2012] and a hand gesture dataset collected by Huazhong University of Science and Technology (Gesture3DMotion). In MSRGesture3D dataset, 10 subjects perform 12 kinds of gestures defined by American Sign Language (ASL), including *bathroom, blue, finish, green, hungry, milk, past, pig, store, where, j* and *z*. This dataset consists of 333 depth sequences. In Gesture3DMotion dataset, 4 subjects perform 12 kinds of rotation hand gestures and contour area changing hand gestures, including *closed rotation, win rotation, one circle, opened rotation, eight rotation, six rotation, closed opened, win transla-*

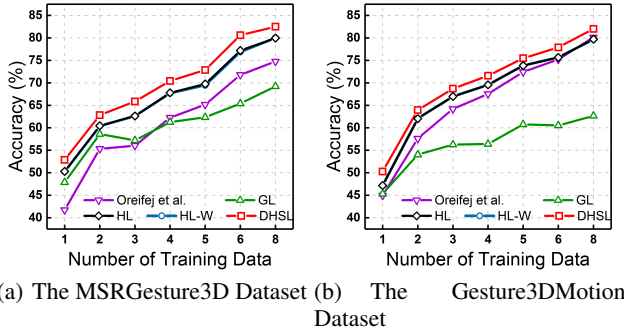


Figure 3: Experimental comparison of different methods on gesture recognition.

tion, one wave, three rotation, one rotation and little rotation. This dataset is composed of 384 depth sequences captured by Kinect-2.

In our experiments, a set of data are selected for training and all other data are used for testing. The training data varies from 1 to 8 for each category respectively. This training/testing data selection process repeats 10 times and the average classification performance is reported. In these experiments, we also empirically set the parameter  $\beta$  and  $\lambda$  as 10 and 1 on both datasets, respectively.

In experiments, the recent gesture recognition method, *i.e.*, HON4D [Oreifej and Liu, 2013] is used for gesture feature extraction method and also for comparison. HON4D is able to capture motion and geometry cues jointly in the 4D space (depth, time, and spatial coordinates). Similar to the 3D shape recognition experiments, we also compare DHSL with GL, HL and HL-W with the HON4D features.

Experimental results on the two datasets are demonstrated in Figure 3. From these results, we can have the following observations:

1. The proposed DHSL outperforms the state-of-the-art method HON4D on both datasets consistently. For example, when 1 and 2 training data are employed for each gesture, the gains are 26.74% and 13.51% on the MSRGesture3D dataset.
2. DHSL performs better than HL in all cases. For example, on the MSRGesture3D dataset, DHSL obtains gains of 15.13%, 5.16% and 5.16% with 3 training data per gesture compared with GL, HL and HL-W, respectively.

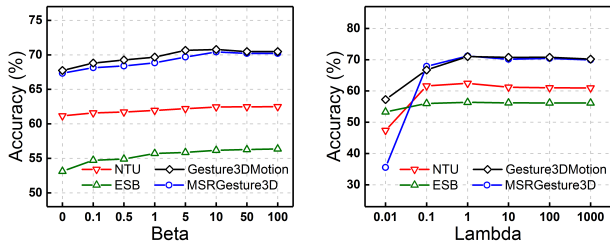


Figure 4: The performance evaluation by varying the parameter  $\beta$  and  $\lambda$  on the four datasets, respectively.

### 4.3 Discussions

In both applications, we can notice that all hypergraph learning methods can achieve better performance compared with the original state-of-the-art methods, *i.e.*, MVCNN and HON4D, which demonstrates the effectiveness of the hypergraph modelling.

Compared with HL, HL-W achieves slightly improvements on the NTU dataset, while is with almost the same performance on the other three datasets (the lines in Fig. 2 and Fig. 3 are overlapped). This phenomenon indicates that the learning of hyperedge weights is not effective on improving the hypergraph structure. Different from HL-W, the proposed dynamic hypergraph structure learning methods can outperform static hypergraph learning methods on two tasks and four datasets consistently.

The better performance of DHSL can be dedicated to two reasons. First, the proposed method introduces a dynamic hypergraph structure to formulate data correlations. Such dynamic hypergraph structure can better fit the data, compared with traditional static hypergraph structure. Second, the hypergraph structure is optimized to be smooth in both the label space and the feature space in DHSL. In this way, the connections among the vertices with the same label and similar features can be more and stronger than those with different labels or dissimilar features. This dynamic hypergraph strategy and the corresponding optimization method can lead to better performance compared with traditional hypergraph learning methods.

### 4.4 On Parameters

In our framework, there are two important parameters that control the relative impact of different components in the objective function, *i.e.*,  $\beta$  and  $\lambda$ , where  $\beta$  is for the smoothness constraint in the feature space and  $\lambda$  is for the empirical loss.

We first keep  $\lambda$  as 1 and vary  $\beta$  from 0 to 100, and evaluate the influence of different  $\beta$  selections on the experimental performance. Experimental results on four datasets are demonstrated in Fig. 4. From these results, we can observe that when  $\beta$  is too small, such as  $\beta = 0$ , the performance is worse than that with a higher  $\beta$  value. When  $\beta$  is too small, the smoothness constraint in the feature space will have little influence on the learning process. When  $\beta$  varies in a large range, such as  $[5, 100]$ , the performance is stable. This result indicates that 1) the use of the smoothness constraint in the feature space is useful to guide the learning of hypergraph structure, and 2) the proposed method is not sensitive to the selection of  $\beta$ .

We then keep  $\beta$  as 10 and vary  $\lambda$  from 0.01 to 1000 to evaluate the influence of different  $\lambda$  selections on the experimental performance. As shown in Fig. 4, when  $\lambda$  is too small, such as  $\lambda = 0.01$ , the performance degenerates significantly. When  $\lambda$  is above 1, the performance becomes stable with respect to the variance of  $\lambda$ . These results indicate that the proposed method is not sensitive to the selection of  $\lambda$ .

## 5 Conclusion

In this paper, we propose the first dynamic hypergraph structure learning method. The proposed method is able to jointly

optimize the hypergraph structure and learn the label projection matrix. For hypergraph structure updating, the data relationships in both the label space and the feature space have been taken into consideration, and the hypergraph structure is dynamically optimized, leading to a better data correlation modelling. We have applied the proposed method on 3D shape recognition and gesture recognition. Experimental results show that the proposed method can achieve better performance compared with the traditional hypergraph learning methods and the state-of-the-art methods. More specifically, the proposed method can obtain consistent gains compared with the traditional hypergraph learning method, indicating the effectiveness of the proposed method on data modelling.

## Acknowledgments

This work was supported by National Key R&D Program of China (Grant No. 2017YFC0113000), National Natural Science Funds of China (U1701262, 61671267), and Tsinghua University Initiative Scientific Research Program (20171080282).

## A Derivations

The derivation of Eq. (9) is as follows:

$$\begin{aligned}\nabla Q(\mathbf{H}) &= \frac{d\text{tr}\left(\left(\mathbf{I} - \mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\right)\mathbf{K}\right)}{d\mathbf{H}} \\ &= -\frac{d\text{tr}\left(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\mathbf{K}\right)}{d\mathbf{H}}\end{aligned}$$

Let

$$\begin{aligned}\mathbf{D}_v^{-\frac{1}{2}} &= \begin{pmatrix} d_1^v & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{n+m}^v \end{pmatrix}, \\ \mathbf{D}_e^{-1} &= \begin{pmatrix} d_1^e & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & d_{n+m}^e \end{pmatrix}.\end{aligned}$$

We obtain

$$\begin{aligned}\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}} &= \begin{pmatrix} d_1^v h_{11} & d_1^v h_{12} & \cdots & d_1^v h_{1l} \\ \vdots & \vdots & \ddots & \vdots \\ d_l^v h_{l1} & d_l^v h_{l2} & \cdots & d_l^v h_{ll} \end{pmatrix} \\ &\quad \begin{pmatrix} w_1 d_1^e & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & w_l d_l^e \end{pmatrix} \begin{pmatrix} d_1^v h_{11} & d_2^v h_{21} & \cdots & d_l^v h_{l1} \\ \vdots & \vdots & \ddots & \vdots \\ d_l^v h_{l1} & d_2^v h_{l2} & \cdots & d_l^v h_{ll} \end{pmatrix} \\ &= \begin{pmatrix} \sum_{j=1}^l w_j d_j^e (d_1^v)^2 h_{1j}^2 & \cdots & \sum_{j=1}^l w_j d_j^e d_1^v h_{1j} h_{lj} \\ \vdots & \ddots & \vdots \\ \sum_{j=1}^l w_j d_j^e d_n^v d_1^v h_{nj} h_{1j} & \cdots & \sum_{j=1}^l w_j d_j^e (d_l^v)^2 h_{lj}^2 \end{pmatrix}.\end{aligned}$$

Therefore,

$$\begin{aligned}\text{tr}\left(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\mathbf{K}\right) &= \sum_{p=1}^l \sum_{i=1}^l k_{pi} \left(\sum_{j=1}^l w_j d_j^e d_i^v d_p^v h_{ij} h_{pj}\right).\end{aligned}$$

Let us also consider the function

$$\begin{aligned}\frac{\partial d_i^v}{\partial h_{st}} &= \frac{\partial \left(\sum_{j=1}^l w_j h_{ij}\right)^{-\frac{1}{2}}}{\partial h_{st}} \\ &= -\frac{1}{2} \left(\sum_{j=1}^l w_j h_{ij}\right)^{-\frac{3}{2}} \sum_{j=1}^l \frac{\partial h_{ij}}{\partial h_{st}} w_j = -\frac{1}{2} (d_i^v)^3 \delta_{is} w_t,\end{aligned}$$

and

$$\begin{aligned}\frac{\partial d_j^e}{\partial h_{st}} &= \frac{\partial \left(\sum_{i=1}^l h_{ij}\right)^{-1}}{\partial h_{st}} \\ &= -\left(\sum_{i=1}^l h_{ij}\right)^{-2} \frac{\partial \sum_{i=1}^l h_{ij}}{\partial h_{st}} = -(d_j^e)^2 \delta_{jt}.\end{aligned}$$

We have

$$\begin{aligned}\frac{\partial \text{tr}\left(\mathbf{D}_v^{-\frac{1}{2}}\mathbf{H}\mathbf{W}\mathbf{D}_e^{-1}\mathbf{H}^T\mathbf{D}_v^{-\frac{1}{2}}\mathbf{K}\right)}{\partial h_{st}} &= \sum_{p=1}^l \sum_{i=1}^l k_{pi} \left(\sum_{j=1}^l \frac{\partial}{\partial h_{st}} (w_j d_j^e d_i^v d_p^v h_{ij} h_{pj})\right) \\ &= \sum_{p,i,j=1}^l k_{pi} w_j \left(\frac{\partial d_j^e}{\partial h_{st}} (d_i^v d_p^v h_{ij} h_{pj}) + d_j^e \frac{\partial d_i^v}{\partial h_{st}} (d_p^v h_{ij} h_{pj})\right) \\ &\quad + d_j^e d_i^v \frac{\partial d_p^v}{\partial h_{st}} (h_{ij} h_{pj}) + d_j^e d_i^v d_p^v \frac{\partial h_{ij}}{\partial h_{st}} h_{pj} + d_j^e d_i^v d_p^v h_{ij} \frac{\partial h_{pj}}{\partial h_{st}} \\ &= \sum_{p,i,j=1}^l k_{pi} w_j \left(\left(\frac{\partial d_i^v}{\partial h_{st}} d_j^e + \frac{\partial d_p^v}{\partial h_{st}} d_i^v\right) (d_p^v h_{ij} h_{pj})\right) \\ &\quad + \left(\frac{\partial h_{ij}}{\partial h_{st}} h_{pj} + \frac{\partial h_{pj}}{\partial h_{st}} h_{ij}\right) d_j^e d_i^v d_p^v + \frac{\partial d_j^e}{\partial h_{st}} (d_i^v d_p^v h_{ij} h_{pj}) \\ &= -\sum_{p=1}^l \sum_{j=1}^l \frac{(k_{ps} + k_{sp})}{2} w_j (d_s^v)^3 w_t d_p^v d_j^e h_{sj} h_{pj} \\ &\quad - \sum_{p,i=1}^l k_{ps} w_t (d_t^e)^2 d_s^v d_p^v h_{st} h_{pt} + \sum_{p=1}^l (k_{ps} + k_{sp}) w_t d_t^e d_s^v d_p^v h_{pt}.\end{aligned}$$

After integrating and simplification, we obtain

$$\begin{aligned}\nabla Q(\mathbf{H}) &= \mathbf{J}(\mathbf{I} \otimes \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{K} \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H}) \mathbf{W} \mathbf{D}_e^{-2} \\ &\quad + \mathbf{D}_v^{-\frac{3}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1} \mathbf{H}^T \mathbf{D}_v^{-\frac{1}{2}} \mathbf{K} \mathbf{J} \mathbf{W} \\ &\quad - 2 \mathbf{D}_v^{-\frac{1}{2}} \mathbf{K} \mathbf{D}_v^{-\frac{1}{2}} \mathbf{H} \mathbf{W} \mathbf{D}_e^{-1}.\end{aligned}$$

## References

- [An *et al.*, 2017] Le An, Xiaojing Chen, Songfan Yang, and Xuelong Li. Person re-identification by multi-hypergraph fusion. *IEEE Transactions on Neural Networks and Learning Systems*, 28(11):2763–2774, 2017.
- [Chen *et al.*, 2003] Ding-Yun Chen, Xiao-Pei Tian, Yu-Te Shen, and Ming Ouhyoung. On visual similarity based 3D model retrieval. *Computer Graphics Forum*, 22(3):223–232, 2003.
- [Du *et al.*, 2017] Dawei Du, Honggang Qi, Longyin Wen, Qi Tian, Qingming Huang, and Siwei Lyu. Geometric hypergraph learning for visual tracking. *IEEE Transactions on Cybernetics*, 47(12):4182–4195, 2017.
- [Gao *et al.*, 2012] Yue Gao, Meng Wang, Dacheng Tao, Rongrong Ji, and Qionghai Dai. 3-D object retrieval and recognition with hypergraph analysis. *IEEE Transactions on Image Processing*, 21(9):4290–4303, 2012.
- [Gao *et al.*, 2013] Yue Gao, Meng Wang, Zheng-Jun Zha, Jialie Shen, Xuelong Li, and Xindong Wu. Visual-textual joint relevance learning for tag-based social image search. *IEEE Transactions on Image Processing*, 22(1):363–376, 2013.
- [Huang *et al.*, 2009] Yuchi Huang, Qingshan Liu, and Dimitris Metaxas. Video object segmentation by hypergraph cut. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 1738–1745, 2009.
- [Huang *et al.*, 2010] Yuchi Huang, Qingshan Liu, Shaoting Zhang, and Dimitris N Metaxas. Image retrieval via probabilistic hypergraph ranking. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 3376–3383, 2010.
- [Hwang *et al.*, 2008] TaeHyun Hwang, Ze Tian, Rui Kuangy, and Jean-Pierre Kocher. Learning on weighted hypergraphs to integrate protein interactions and gene expressions for cancer outcome prediction. In *Proceedings of IEEE International Conference on Data Mining*, pages 293–302, 2008.
- [Jayanti *et al.*, 2006] Subramaniam Jayanti, Yagnanarayanan Kalyanaraman, Natraj Iyer, and Karthik Ramani. Developing an engineering shape benchmark for cad models. *Computer-Aided Design*, 38(9):939 – 953, 2006.
- [Li and Milenkovic, 2017] Pan Li and Olgica Milenkovic. Inhomogeneous hypergraph clustering with applications. In *Proceedings of Advances in Neural Information Processing Systems*, pages 2305–2315. Curran Associates, Inc., 2017.
- [Liu *et al.*, 2017a] Mingxia Liu, Jun Zhang, Pew-Thian Yap, and Dinggang Shen. View-aligned hypergraph learning for Alzheimer’s disease diagnosis with incomplete multimodality data. *Medical Image Analysis*, 36:123–134, 2017.
- [Liu *et al.*, 2017b] Qingshan Liu, Yubao Sun, Cantian Wang, Tongliang Liu, and Dacheng Tao. Elastic net hypergraph learning for image clustering and semi-supervised classification. *IEEE Transactions on Image Processing*, 26(1):452–463, 2017.
- [Luo *et al.*, 2018] Fulin Luo, Bo Du, Liangpei Zhang, Lefei Zhang, and Dacheng Tao. Feature learning using spatial-spectral hypergraph discriminant analysis for hyperspectral image. *IEEE Transactions on Cybernetics*, 2018.
- [Oreifej and Liu, 2013] Omar Oreifej and Zicheng Liu. Hon4d: Histogram of oriented 4D normals for activity recognition from depth sequences. In *Proceedings of IEEE Conference on Computer Vision and Pattern Recognition*, pages 716–723, 2013.
- [Su *et al.*, 2015] Hang Su, Subhransu Maji, Evangelos Kalogerakis, and Erik Learned-Miller. Multi-view convolutional neural networks for 3D shape recognition. In *Proceedings of IEEE International Conference on Computer Vision*, 2015.
- [Wang *et al.*, 2012] Jiang Wang, Zicheng Liu, Jan Chorowski, Zhuoyuan Chen, and Ying Wu. Robust 3D action recognition with random occupancy patterns. In *Proceedings of European Conference on Computer Vision*, pages 872–885, 2012.
- [Zhang *et al.*, 2017] Zhihong Zhang, Lu Bai, Yuanheng Liang, and Edwin Hancock. Joint hypergraph learning and sparse regression for feature selection. *Pattern Recognition*, 63:291 – 309, 2017.
- [Zhao *et al.*, 2018] Wei Zhao, Shulong Tan, Ziyu Guan, Boxuan Zhang, Maoguo Gong, Zhengwen Cao, and Quan Wang. Learning to map social network users by unified manifold alignment on hypergraph. *IEEE Transactions on Neural Networks and Learning Systems*, 2018.
- [Zhou and Schölkopf, 2007] Jiayuan Huang Zhou, Denny and Bernhard Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Proceedings of Advances in Neural Information Processing Systems*, pages 1601–1608. MIT Press, 2007.
- [Zhou *et al.*, 2003] Denny Zhou, Olivier Bousquet, Thomas N Lal, Jason Weston, and Bernhard Schölkopf. Learning with local and global consistency. In *Proceedings of Advances in Neural Information Processing Systems*, pages 321–328, 2003.
- [Zhu *et al.*, 2015] Lei Zhu, Jialie Shen, Hai Jin, Ran Zheng, and Liang Xie. Content-based visual landmark search via multimodal hypergraph learning. *IEEE Transactions on Cybernetics*, 45(12):2756–2769, 2015.
- [Zhu *et al.*, 2017a] Lei Zhu, Jialie Shen, Liang Xie, and Zhiyong Cheng. Unsupervised topic hypergraph hashing for efficient mobile image retrieval. *IEEE transactions on cybernetics*, 47(11):3941–3954, 2017.
- [Zhu *et al.*, 2017b] Xiaofeng Zhu, Yonghua Zhu, Shichao Zhang, Rongyao Hu, and Wei He. Adaptive hypergraph learning for unsupervised feature selection. In *Proceedings of International Joint Conference on Artificial Intelligence*, pages 3581–3587, 2017.