Hindawi
Computational Intelligence and Neuroscience
Volume 2022, Article ID 9073652, 13 pages
https://doi.org/10.1155/2022/9073652

Research Article

Multimodal Feature Fusion Based Hypergraph Learning Model

Zhe Yang, [1,2,3] **Liangkui Xu**, [1,2,3] **and Lei Zhao** [1,2,3]

[1] *School of Computer Science & Technology, Soochow University, Suzhou, China*
[2] *Provincial Key Laboratory for Computer Information Processing Technology, Suzhou, China*
[3] *Provincial Key Laboratory for Intelligent Engineering in Big Data, Suzhou, China*

Correspondence should be addressed to Zhe Yang; yangzhe@suda.edu.cn

Received 15 March 2022; Accepted 25 April 2022; Published 16 May 2022

Academic Editor: Kapil Sharma

Hypergraph learning is a new research hotspot in the machine learning field. The performance of the hypergraph learning model depends on the quality of the hypergraph structure built by different feature extraction methods as well as its incidence matrix. However, the existing models are all hypergraph structures built based on one feature extraction method, with limited feature extraction and abstract expression ability. This paper proposed a multimodal feature fusion method, which firstly built a single modal hypergraph structure based on different feature extraction methods, and then extended the hypergraph incidence matrix and weight matrix of different modals. The extended matrices fuse the multimodal abstract feature and an expanded Markov random walk range during model learning, with stronger feature expression ability. However, the extended multimodal incidence matrix has a high scale and high computational cost. Therefore, the Laplacian matrix fusion method was proposed, which performed Laplacian matrix transformation on the incidence matrix and weight matrix of every model, respectively, and then conducted a weighted superposition on these Laplacian matrices for subsequent model training. The tests on four different types of datasets indicate that the hypergraph learning model obtained after multimodal feature fusion has a better classification performance than the single modal model. After Laplace matrix fusion, the average time can be reduced by about 40% compared with the extended incidence matrix, the classification performance can be further improved, and the index F1 can be improved by 8.4%.

1. Introduction

In the machine learning field, the graph is an important data model. If the research objects have a one-to-one relationship between each other, then they can be solved by an ordinary graph such as social networks, gene data, and web page ranking problems [1]. However, in reality, the objects always have a complicated one-to-many or many-to-many relationship between each other [2]. Taking a reference citation, for example, a thesis can cite multiple papers and can be cited by multiple papers. When solving with ordinary graphs, the multivariate relationship will be forcibly shifted into a binary relationship, simply causing information loss. Thus, a hypergraph–a variant of ordinary graph–emerges [3]. Since hypergraphs can better describe the multivariate information between objects, in recent years, hypergraph-based machine learning has become a research hotspot of the machine learning field and has obtained good effect in object segmentation [4], disease diagnosis [5], image classification [6], recommendation system [2, 7], etc.

There are two main methods of using the hypergraph learning model to solve the multivariate relation problem. One method is to extend the hypergraph into an ordinary graph and then use the ordinary graph to solve the hypergraph problems. The representative methods include clique extension, star extension, and line extension [8]. However, in the process of extending a hypergraph to an ordinary graph, the multivariate relation between vertices is changed into a binary relation, which may cause information loss. The other method is to directly aim at the hypergraph structure and its incidence matrix, and solve the optimal hypergraph cut after Laplacian matrix transformation, that is to say, obtain several tangent vectors of the hypergraph Laplacian matrix, and divide the hypergraph into different subsets for classification and clustering. In essence, this is a combinational optimization problem, and the representative methods include the

Zhou's normalized Laplacian [3], hypergraph learning regularity optimization [6, 9, 10], hypergraph multimodal structure [11], and hypergraph deep learning [12]. Such methods directly carry out Laplacian transformation and solving on the hypergraph incidence matrix, preventing information loss due to structural transformation. However, all of them are based on one feature extraction method to build a single modal hypergraph structure. If the feature extraction method is not capable enough to fully reflect the relation between objects, it will lead to low-quality hypergraph building and its incidence matrix, finally affecting the performance of the hypergraph learning model.

Therefore, this paper proposed a multimodal feature fusion method, using incidence matrix extension and Laplacian matrix fusion to improve the model performance. The specific work and innovations include the following conditions.

(1) When building a hypergraph, first, a single modal hypergraph structure is built based on different feature extraction methods, then different models of hypergraph Laplacian matrices and their weight matrices are extended, and the extended multimodal Laplacian matrices receive model training. The test results show that the extended multimodal incidence matrix could effectively improve the classification performance of the hypergraph learning model.

(2) If the dimension of the multimodal incidence matrix obtained by matrix extension is high, it will lead to high computational cost. Therefore, the Laplacian matrix fusion method is put forward, which firstly performs Laplacian matrix transformation on the hypergraph incidence matrix and weight matrix of every model, respectively, and then, these Laplacian matrices are weighted accumulated for further model training. The test results indicate that the Laplacian matrix fusion method can not only reduce the computational cost of the multimodal incidence matrix but also improve the model performance.

2. Related Work

Since Zhou et al. firstly proposed the hypergraph learning model [3] and used the "Markov random walk" idea to explain the model, in recent years, there are mainly two approaches to solving multivariate relation problems with the hypergraph learning model: extension method and segmentation method.

The extension method is to expand the hypergraph into an ordinary graph and then use the ordinary graph method to solve the hypergraph problem. Zien firstly raised the star extension approach [13], introducing a new node to every hyperedge of hypergraphs, in which the new node is connected to every vertex of hyperedge by an edge. This approach does not take into account the connection relationship between vertexes within the same hyperedge. Afterwards, Agarwal puts forward a clique extension approach [14], considering every hyperedge as a fully connected subgraph. Although the relationship between vertexes within the same hyperedge is built, it still cannot comprehensively express the connection information of vertexes between hyperedges [12, 15]. To further reduce the information loss during extension, Yang et al. proposed a line extension approach [8] in 2020, considering hyperedges and the nodes within every hyperedge as a new vertex. This approach maximally retains the connection information of vertexes and hyperedges, but it may still bring information loss when dealing with some problems, such as the Fano plane problem [16]. To avoid the information loss problem led by structural transformation, Gao introduced the multimodal concept [11], making different modals correspond to one subhypergraph with weights and training parameters for all subhypergraphs. However, this approach has too many parameters to be optimized, resulting in a high time cost for training. In 2019, Feng proposed the hypergraph Laplacian extension method, which optimized the hypergraph structure while reducing the time cost of machine learning [17]. However, this extension method merely extends the incidence matrix without considering the impact of the weight matrix on the model.

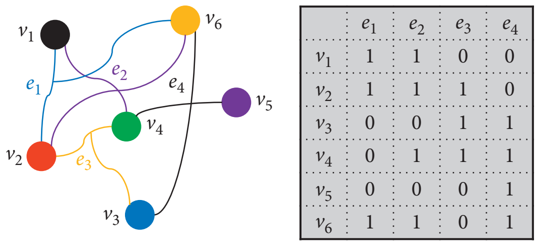
The segmentation method is mainly based on the hypergraph Laplacian matrix to solve the optimal hypergraph cut. Chen et al. applied L2 regularization to optimize the weight parameters of the hypergraph learning model [6, 9]. Chen and Luo used the alternating least square (ALS) method [9] and the coordinate descent method [10] to optimize weight parameters, respectively. Guo utilized the random matrix diffusion idea to optimize the hypergraph Laplacian optimal cut, but this was limited by a single hypergraph structure, needed to optimize the additional target function, and had too high time cost [18]. After that, Zhang et al. raised a hypergraph inductive learning model using the category projection matrix to obtain the category label of the sample [19]. Although the time complexity was brought down to a certain extent, not all dataset information was used in model training, so the classification performance was lowered slightly.

Hence, all existing hypergraph model's solving methods have certain restrictions. The extension approach breaks the multivariate relation between objects. The segmentation approach does not break the limitation of a single hypergraph structure but only optimizes the model solving process, easy to bring high time cost. The multimodal feature fusion method proposed in this paper fuses the abstract features of multiple modals by matrix extension and breaks the limitation of a single hypergraph structure. And in model solving, the Laplacian matrix fusion method helps to reduce the time cost and further enhance the model performance.

3. Multimodal Feature Fusion Based Hypergraph Model

3.1. General Hypergraph Model. Let $G = (V, E, W)$ denote a hypergraph, where $V = \{v_1, v_2, \ldots v_n,\}$ is the vertex set, and $E = \{e_1, e_2, \ldots e_m,\}$ is the hyperedge set. The hypergraph's every hyperedge $e_j (1 \leq j \leq m)$ contains multiple vertexes $v_i (1 \leq i \leq n)$, so the hypergraph G can be expressed by incidence matrix $H \in R^{n \times m}$, as shown in Figure 1. In H, every element $h(v_i, e_j)$ is defined as

$$h(v_i, e_j) = \begin{cases} 1, & \text{if } v_i \in e_j, \\ 0, & \text{if } v_i \notin e_j. \end{cases} \tag{1}$$



	e_1	e_2	e_3	e_4
v_1	1	1	0	0
v_2	1	1	1	0
v_3	0	0	1	1
v_4	0	1	1	1
v_5	0	0	0	1
v_6	1	1	0	1

(a) (b)

Figure 1: Hypergraph and incidence matrix. (a) Hypergraph. (b) Incidence matrix.

Every hyperedge e_j is given a positive weight $w(e_j)$, and $w(e_j)$ is calculated by the Gaussian kernel method [20]. Where $dis(v_i, v_k)(1 \le i, k \le n)$ represents the Euclidean distance between vertex v_i and v_k, and σ represents the average of the distances between all vertexes.

$$w(e_j) = \sum_{v_i, v_k \in e_j} \exp\left(-\frac{\operatorname{dis}(v_i, v_k)^2}{\sigma^2}\right). \tag{2}$$

The diagonal matrix $W \in R^{m\times m}$ is defined, and the elements in diagonal are the weight of every hyperedge $w(e_j)(1 \le j \le m)$, as shown in the following equation:

$$W = \begin{bmatrix} w(e_1) & 0 & \cdots & 0 \\ 0 & w(e_2) & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & 0 & w(e_m) \end{bmatrix}. \tag{3}$$

In the hypergraph, the degree $d(v_i)$ of every vertex v_i is defined as shown in the following equation:

$$d(v_i) = \sum_{j=1}^{m} w(e_j) h(v_i, e_j). \tag{4}$$

The degree $\delta(e_j)$ of hyperedge e_j is defined as the number of vertexes included in this hyperedge, as shown in the following equation:

$$\delta(e_j) = \sum_{i=1}^{n} h(v_i, e_j). \tag{5}$$

Two diagonal matrix $D_v \in R^{n\times n}$ and $D_e \in R^{m\times m}$ is defined. Similar to equation (3), their diagonal elements are the degree of every vertex $d(v_i)$ and the degree of every hyperedge $\delta(e_j)$ in the hypergraph.

The regularity optimization target function of the hypergraph learning model is shown in the following equation [6, 9]:

$$\arg\min_{F} \Phi(F) \coloneqq \Omega(F) + \lambda\theta(F) + \mu\|W\|^2, \tag{6}$$

where λ and μ are regularization parameters, to balance each item of equation (6). For a c classification problem, $F \in R^{n\times c}$ is the eventually solved tangent vector matrix, including the predicted sample category information. $Y \in R^{n\times c}$ is the vector containing real sample labels. In case the vertex v_i falls into the $k(0 \le k \le c)$ category, then $Y(v_i)[k] = 1$, $Y(v_i)[p] = 0(0 \le p \le c\; p \ne k)$. $\theta(F)$ is the experience loss function, and $\Omega(F)$ is the standard loss function, see definitions in the following equations:

$$\theta(F) = \|Y - F\|_2^2, \tag{7}$$

$$\begin{aligned}
\Omega(F),& \\
&= \frac{1}{2}\sum_{\substack{j=1 \\ v_i, v_k \in e_j}}^{m} \frac{w(e_j)h(v_i, e_j)h(v_k, e_j)}{\delta(e_j)}\left(\frac{F(v_i)}{\sqrt{d(v_i)}} - \frac{F(v_k)}{\sqrt{d(v_k)}}\right)^2, \\
&= \sum_{\substack{j=1 \\ v_i, v_k \in e_j}}^{m} \frac{w(e_j)h(v_i, e_j)h(v_k, e_j)}{\delta(e_j)}\left(\frac{F^2(v_i)}{d(v_i)} - \frac{F(v_i)F(v_k)}{\sqrt{d(v_i)d(v_k)}}\right), \\
&= \sum_{k=1}^{n} F^2(v_k) \sum_{j=1}^{m} \frac{w(e_j)h(v_i, e_j)}{d(v_i)} \sum_{i=1}^{n} \frac{h(v_k, e_j)}{\delta(e_j)} - \sum_{\substack{j=1 \\ v_i, v_k \in e_j}}^{m} \sum \frac{F(v_i)h(v_i, e_j)w(e_j)h(v_k, e_j)F(v_k)}{\sqrt{d(v_i)d(v_k)}\delta(e_j)}, \\
&= F^T\left(I - D_v^{-1/2}\mathrm{HWD}_e^{-1}H^T D_v^{-1/2}\right)F, \\
&= F^T LF.
\end{aligned} \tag{8}$$

In equation (8), L is the hypergraph standardization Laplacian matrix.

$$L = I - D_v^{-1/2}\mathrm{HWD}_e^{-1}H^T D_v^{-1/2}. \tag{9}$$

In equation (8), only if two vertexes v_i and $v_k\,(k \neq i)$ are at the same hyperedge, it is required to make their standardized labels $F(v_i)/d(v_i)$ and $F(v_k)/d(v_k)$ to be similar as possible, so that the vertex label can be predicted more accurately. If the hypergraph has degenerated to ordinary graph, then D_e is degenerated to $2I$, and it is obtained from equation (10) that the relationship between hypergraph's Laplacian matrix L and ordinary graph's Laplacian matrix L_A is 1/2 coefficient, in which A represents the adjacent matrix of an ordinary graph.

$$\begin{aligned} L &= I - D_v^{-1/2}\mathrm{HWD}_e^{-1}H^T D_v^{-1/2},\\ &= I - \frac{1}{2}D_v^{-1/2}\mathrm{HWH}^T D_v^{-1/2},\\ &= I - \frac{1}{2}D_v^{-1/2}\left(D_v + A\right)D_v^{-1/2},\\ &= \frac{1}{2}\left(I - D_v^{-1/2}AD_v^{-1/2}\right),\\ &= \frac{1}{2}L_A. \end{aligned} \tag{10}$$

In equation (6), the variables in the optimization target function to be determined are W and F. Because the target function (6) alone is a convex function relative to W and F, so it is feasible to use the ALS method [9] and coordinate descend method [10] to optimize W and F.

In the case of using ALS, it is first to fix W, let $\delta\Phi(F)/\delta F = 0$, then F is optimized.

$$F = \left(I + \frac{1}{\lambda}L\right)^{-1}Y. \tag{11}$$

Second, fix F and optimize W. By letting $\delta\Phi(W)/\delta W = 0$, W can be updated as

$$W = \max\left(\frac{F^T D_v^{-1/2}HI_\theta D_e^{-1}H^T D_v^{-1/2}F}{2\mu}, 0\right). \tag{12}$$

Third, repeat the above two steps until W and F tend to be stable.

In the case of using the coordinate descending method, in each iteration process, two values W_j and W_k should be selected from W for updating, and such updating shall be conducted under the constraint conditions of (13), while limiting $\sum_{i=0}^{n} W_i = 1$. W_j^* and W_k^* are the values of updated W_j and W_k, see updating process in (14). When W becomes stable, we finally obtain an F, that is, the ultimate label vector, and the category of a vertex v_i corresponds to $\mathrm{index}(\max(F(v_i)))$, that is, the subscript of $F(v_i)$ maximum.

$$\begin{cases} \left[r_1, r_2, \ldots r_n\right] = F^T D_v^{-1/2}H,\\ g_i = -r_i^2 D_e^{-1}(i,i), \end{cases} \tag{13}$$

$$\begin{cases} \begin{aligned} W_j^* &= 0\\ W_k^* &= W_j + W_k \end{aligned} & ,\text{if } 2\mu\left(W_j + W_k\right) + \left(g_k - g_j\right) \leq 0,\\ \begin{aligned} W_j^* &= W_j + W_k\\ W_k^* &= 0 \end{aligned} & ,\text{if } 2\mu\left(W_j + W_k\right) + \left(g_j - g_k\right) \leq 0,\\ \begin{aligned} W_j^* &= \frac{2\mu\left(W_j + W_k\right) + \left(g_k - g_j\right)}{4\mu}\\ W_k^* &= W_j + W_k - W_j^*. \end{aligned} & ,\text{otherwise}, \end{cases} \tag{14}$$

3.2. Hypergraph Laplacian Matrix. The N classification problem solved by the hypergraph model is an N-path hypergraph cut problem in essence while the solution to N-path hypergraph cut is actually the N feature vectors [3] of hypergraph Laplacian matrix. Therefore, if Laplacian matrix can describe more globally and in-depth hypergraph structural information, the accuracy of the hypergraph cut will be increased, and the corresponding classifying performance will also be improved. Zhou et al. firstly proposed the hypergraph learning model [1], adopted Markov random walk idea, and analyzed the hypergraph Laplacian matrix from a probability perspective. The results indicate that the target function (6) of the hypergraph learning model is derived through Markov random walk process. The Markov random walk probability of every vertex in the hypergraph shall follow the following rule: given the current position as $v_i \in V$, first, choose a hyperedge e_j from all hyperedges relevant to the vertex v_i at a certain probability and then uniformly randomly choose a vertex $v_k \in e_j\,(k \neq i)$. Equation (15) provides the probability of a hypergraph vertex's Markov random walk, and equation (16) is a matrix expression of the calculation process of equation (15).

$$p\left(v_i, v_k\right) = \sum_{j=1}^{m} w\left(e_j\right)\frac{h\left(v_i, e_j\right)}{\delta\left(e_j\right)}\frac{h\left(v_k, e_j\right)}{d\left(v_i\right)}, \tag{15}$$

$$P = D_v^{-1/2}\mathrm{HWD}_e^{-1}H^T D_v^{-1/2}. \tag{16}$$

Let S denotes a subset of hypergraph vertex set V, S^c is the complementary set of S, and a cut of a hypergraph G is dividing a hypergraph into two parts S and S^c. VolS represents the volume of the set S, and ∂S is the set of cut hyperedge, $\partial S = \left\{e_j \in E | e_j IS \neq \phi, e_j \cap S^c \neq \phi\right\}$. Equations (17) and (18) define the calculation method of Vol$S$ and Vol∂S.

$$\mathrm{VolS} = \sum_{v_k \in S} d\left(v_k\right), \tag{17}$$

$$\mathrm{VolS}\ \partial\mathrm{S} = \sum_{e_j \in \partial S} w(e)\frac{\left|e_j \cap S\right|\left|e_j \cap \partial S\right|}{\delta\left(e_j\right)}. \tag{18}$$

Equations (19)–(20) prove the stationary distribution of Markov random walk as $\pi(v_k)$ [1].

$$\pi(v_k) = \frac{d(v_k)}{\mathrm{VolV}}, \tag{19}$$

$$\begin{aligned}
&\sum_{i=1}^{n} \pi(v_i) p(v_i, v_k),\\
&= \sum_{i=1}^{n} \frac{d(v_i)}{\mathrm{VolV}} \sum_{j=1}^{m} w(e_j) \frac{h(v_i, e_j)}{d(v_i)},\\
&= \frac{1}{\mathrm{VolV}} \sum_{i=1,j=1}^{n,m} \frac{w(e_j) h(v_i, e_j) h(v_k, e_j)}{\delta(e_j)},\\
&= \frac{1}{\mathrm{VolV}} \sum_{j=1}^{m} w(e_j) \sum_{i=1}^{n} h(v_i, e_j) \frac{h(v_k, e_j)}{\delta(e_j)},\\
&= \frac{1}{\mathrm{VolV}} \sum_{j=1}^{m} w(e_j) h(v_k, e_j),\\
&= \frac{d(v_k)}{\mathrm{VolV}}.
\end{aligned} \tag{20}$$

It can be seen from equations (15)–(16) that the process of Markov random walk conforms to the hypergraph's standard Laplacian operator L in equation (9). If V is the feature vector of P, then V is also the feature vector of L. The Laplacian matrix can be deemed as the expression of the hypergraph structure after a random walk. The target function of the hypergraph cut is (21), and equations (19)–(23) prove the minimal cut of Markov random walk, that is to say, making the similarity of the edge connecting different clusters to be minimal (the vertex transition probability is the minimum), while the vertex transition probability within the same cluster is maximal and gradually tends to a stably distributed partition. On account of the difficulty in directly solving (21), it is converted to the basic loss function (8) corresponding to (6), that is, $\arg\min \Omega(F)$. The solution to $\arg\min \Omega(F)$ is the feature vector corresponding to N minimal nonzero eigenvalues of hypergraph Laplacian matrix. Thus, the Laplacian matrix is very critical to the solving of the hypergraph problem.

$$\begin{aligned}
&\min \mathrm{NCut}(S, S^c),\\
&= \min \frac{\mathrm{Vol}\partial S}{\mathrm{VolV}} \left(\frac{1}{\mathrm{VolS}/\mathrm{VolV}} + \frac{1}{\mathrm{VolS}^c/\mathrm{VolV}} \right),\\
&= \min \mathrm{Vol}\partial S \left(\frac{1}{\mathrm{VolS}} + \frac{1}{\mathrm{VolS}^c} \right).
\end{aligned} \tag{21}$$

$$\frac{\mathrm{VolS}}{\mathrm{VolV}} = \sum_{v_k \in \partial S} \frac{d(v_k)}{\mathrm{VolV}} = \sum_{v_k \in V} \pi(v_k), \tag{22}$$

$$\begin{aligned}
&\frac{\mathrm{Vol}\partial S}{\mathrm{VolV}},\\
&= \sum_{e_j \in \partial S} \frac{w(e_j)}{\mathrm{VolV}} \frac{|e_j \cap S||e_j \cap S^c|}{\delta(e_j)},\\
&= \sum_{e_j \in \partial S} \sum_{v_i \in e_j \cap S} \sum_{v_k \in e_j \cap S^c} \frac{w(e_j)}{\mathrm{Vol}V} \frac{h(v_i, e_j) h(v_k, e_j)}{\delta(e_j)},\\
&= \sum_{e_j \in \partial S} \sum_{v_i \in e_j \cap S} \sum_{v_k \in e_j \cap S^c} w(e_j) \frac{d(v_i)}{\mathrm{VolV}} \frac{h(v_i, e_j)}{d(v_i)} \frac{h(v_k, e_j)}{\delta(e_j)},\\
&= \sum_{v_i \in S} \sum_{v_i \in S^c} \frac{d(v_i)}{\mathrm{Vol}V} \sum_{e_j \in S} w(e) \frac{h(v_i, e_j)}{d(v_i)} \frac{h(v_k, e_j)}{\delta(e_j)},\\
&= \sum_{v_i \in S} \sum_{v_i \in S^c} \pi(v_i) P(v_i, v_k).
\end{aligned} \tag{23}$$

3.3. Incidence Matrix Extension. According to equation (9), the hypergraph Laplacian matrix is relevant to the hypergraph incidence matrix H and weight matrix W. Generally, in the construction of hypergraph structure, it is first to extract the vectorization feature of objects, then similar vertexes are connected in the same one hyperedge based on vertex similarity, and finally, the point-edge incidence matrix H and weight matrix W are used to represent the basic structure of the hypergraph model. The reason that hypergraphs can depict more information than ordinary graphs is the Laplacian matrices H and I express more relationships between vertices and vertices and between vertices and hyperedges.

According to equation (15), the probability of every vertex's random walk in the hypergraph is calculated based on the hypergraph incidence matrix. In this process, the hypergraph vertexes can fuse more neighborhood information to improve the vertex classification accuracy. Markov random walk is a conductive, reasonable incidence matrix extension that not only makes Markov random walk to contain vertexes' neighborhood information in the process but also has the opportunity to contact farther vertexes for exploring global information, finally obtain a hypergraph Laplacian matrix fusing more global information. After blending different Laplacian matrices H together, the originally nonconnected two vertexes are connected again by different short-path combinations, which can achieve the effect of random matrix P "diffusion mapping" [19]. Meanwhile, it is possible to discover the geometry with different scales in heterogeneous hypergraph space, and compared to the original space, it can also keep the globality of hypergraph geometry.

Because different feature extraction methods obtain different feature spaces with different corresponding hypergraph structures, so the incidence matrix extension can be considered as the fusion of multimodal feature space.

As shown in Figure 2, a single modal hypergraph incidence matrix $H_i \in R^{n\times m}$ is built based on the different feature extraction methods, and then, H_i is further blended to obtain the multimodal incidence matrix $H' = [H_1 \| H_2 \cdots \| H_n] \in R^{k\times (n\times m)}$. Hence, the hypergraph vertex probability transition formula can be rewritten as (24), with its matrix expression means as shown in the following equation:

$$p(v_i, v_k) = \sum_{t=1}^{n}\sum_{j=1}^{m} w(e_j)\frac{h_t(v_i, e_j)}{\delta(e_j)}\frac{h_t(v_k, e_j)}{d(v_i)},$$

$$P' = D_v'^{-1/2} H' W' D_e'^{-1} H'^{T} D_v'^{-1/2}. \tag{24}$$

It can be seen from equation (2) that the calculation of hypergraph weight is mainly affected by intervertex distance, and the sample feature extraction method or intersample similarity calculation method is different, and then, the intervertex distance obtained will be different, so the weight matrix $W_i \in R^{m\times m}$ obtained from different modals of incidence matrix H_i will be different. To fuse the hyperedge weight information, the different models of weight matrix W_i are also blended to obtain the multimodal weight matrix $W' = [W_1 \| W_2 \cdots \|| W_n] \in R^{k\times (m\times m)}$. As shown in Figure 2, substitute H' and W' into (9) to obtain $L' = I - D_v^{-1/2} H' W' D_e^{-1} H'^{T} D_v^{-1/2}$. L' is substituted into target function (6), and the gradient descend method is used to solve. From the test data in Table 3, the matrix extension method can improve the model classification performance to a certain extent. This indicates that the extended multimodal incidence matrix and weight matrix fuse more abstract features, and the corresponding Laplacian matrix L' contains richer information, so the hypergraph cut obtained is more accurate with a better classification effect. However, as matrix extension results in the rapid growth of matrix dimension, W' and H' are expanded k times than original matrices W and H, bringing greater time cost. In the next section, a Laplacian matrix fusion method will be proposed to solve this problem.

3.4. Laplacian Matrix Fusion. Figure 3 describes the main process of Laplacian matrix fusion. Different modals of hypergraph structures correspond to different Laplacian matrices. On the basis of building every modal hypergraph incidence matrix H_i and weight matrix W_i, we firstly figure out the corresponding Laplacian matrix L_i under each modal and later perform a weighted sum denoted as L''; that is,

$$L'' = \sum_{i=1}^{K} L_i \cdot r, s.t. r = \frac{1}{K}, \tag{25}$$

K represents the number of fusion Laplacian matrices, and then, the standardized loss function $\Omega(F)$ is rewritten as

$$\begin{aligned}
&\Omega(F),\\
&= \frac{r}{2}\sum_{\substack{g=1,j=1\\ v_i,v_u \in e_j}}^{K,m} \frac{w_g(e_j)h_g(v_i,e_j)h_g(v_u,e_j)}{\delta_g(e_j)}\left(\frac{F(v_i)}{\sqrt{d_g(v_i)}} - \frac{F(v_u)}{\sqrt{d_g(v_u)}}\right)^2,\\
&= r\sum_{\substack{g=1,j=1\\ v_i,v_u \in e_j}}^{K,m} \frac{w_g(e_j)h_g(v_i,e_j)h_g(v_u,e_j)}{\delta_g(e_j)}\left(\frac{F^2(v_i)}{d_g(v_i)} - \frac{F(v_i)F(v_u)}{\sqrt{d_g(v_i)d_g(v_u)}}\right),\\
&= r\sum_{g=1}^{K}\left(\sum_{u=1}^{n} F^2(v_k)\sum_{j=1}^{m}\frac{w_g(e_j)h_g(v_i,e_j)}{d_g(v_i)}\sum_{i=1}^{n}\frac{h_g(v_u,e_j)}{\delta_g(e_j)} - r\sum_{j=1}^{m}\sum_{v_i,v_u\in e_j}\frac{F(v_i)h_g(v_i,e_j)w_g(e_j)h_g(v_u,e_j)F(v_u)}{\sqrt{d_g(v_i)d_g(v_u)}\,\delta_g(e_j)}\right),\\
&= F^T\left(\sum_{g=1}^{K}\left(I - D_{gv}^{-1/2}H_g W_g D_{eg}^{-1}H_g^T D_{gv}^{-1/2}\right)\cdot r\right)F,\\
&= F^T L'' F.
\end{aligned} \tag{26}$$

In Figure 3, the matrix L_i obtained based on each modal hypergraph contains the information of a single modal incidence matrix H_i and weight matrix W_i while the matrix L'' fuses the matrix L_i under all modals, containing more comprehensive and higher quality information. Same as Section 3.3, the fused matrix L'' uses the gradient descent

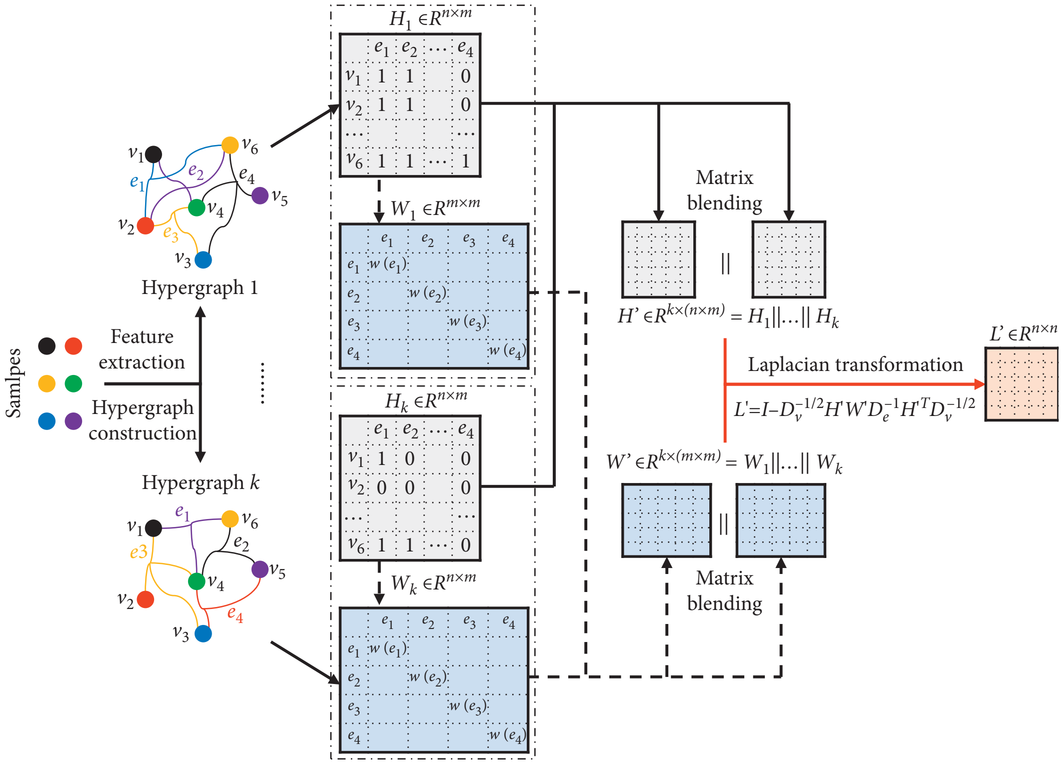


Figure 2: Hypergraph matrix extension (denotes blend operation).

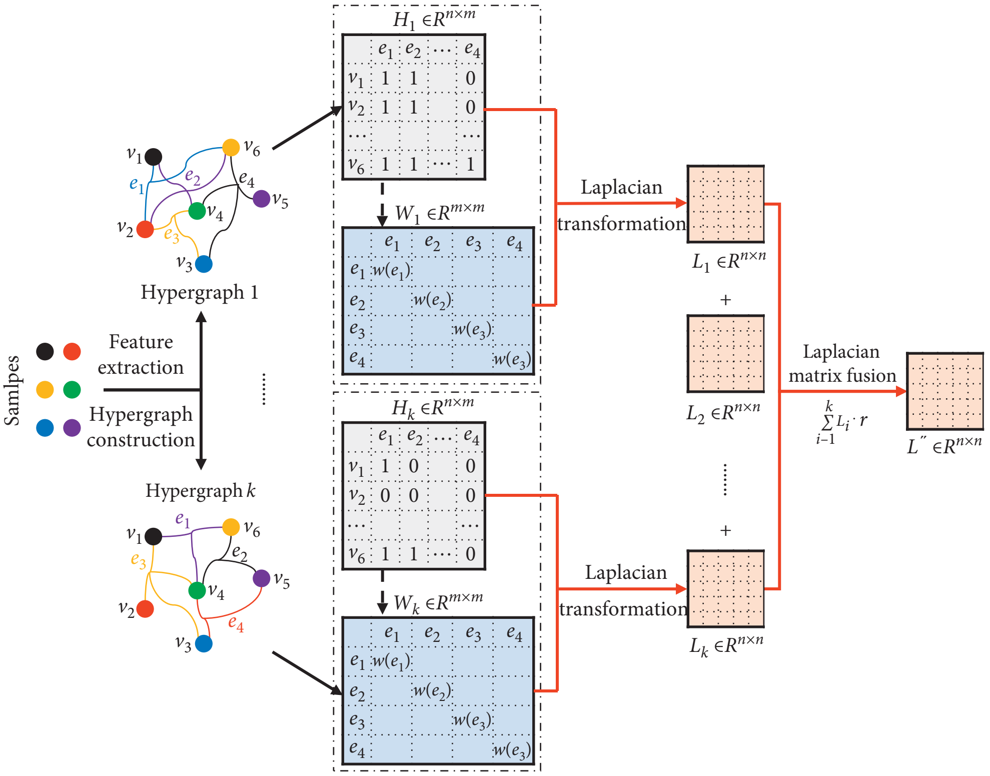


Figure 3: Laplacian matrix fusion.

method to solve the target function (6) and trains the ultimate model.

4. Experiment and Analysis

4.1. Experimental Environment. Aiming at the classification problem, the experiment in this paper compared the hypergraph learning model obtained by training with the proposed method with a typical classification model, ordinary graph, and other hypergraph models. The experimental environment is as follows.

4.1.1. Hardware. The experiment was carried out on GPU colony, available resources as CPU 96 Core, GPU 2 Core GeForce_RTX_2080_Ti/2 Core Tesla_v100_ sxM2_32 GB, memory 512 GB, memory space 500 GB. The programming language is Python 3.7.

4.1.2. Parameters. The KNN method was used to build a hyperedge [21], and that is to say, the vertexes included in the hyperedge are the central vertex and K vertexes nearest to it. To ensure variable consistency, the regularization parameters λ and μ are set as 2; according to the best result of the experiment, the number of vertices contained in the hyperedge is set to 25.

4.1.3. Datasets. To better verify the performance of the proposed model, four different fields of datasets were selected for the experiment, as shown in Table 1.

Cat & Dog. Image datasets, sourced from the official website of Kaggle [22], containing images of cats and dogs, are often used for classification tasks.

Cifar 10. Image dataset, a typical computer vision dataset for object identification and classification [23], included a total of 10 categories.

Ctrip. Text datasets used the hotel comment data of CTRIP in 2018 [24], in which every comment has been labeled with an emotional direction, such as positive comments or negative comments.

Spambase. Numerical value data, the spam datasets provided by the official website of UCI [25], is mainly used to identify and classify spam.

4.1.4. Evaluation Metrics. In this paper, the evaluation indexes used are accuracy, precision, recall, and F1, see the calculation formula in (28)–(30). TP represents the number of samples that are actually positive and predicted as positive, FP represents the number of samples that are actually negative but predicted as positive, FN represents the number of samples that are actually positive but predicted as negative, TN represents the number of samples that are actually negative and predicted as negative.

Accuracy refers to the ratio of the prediction samples correctly classified in total samples.

$$\text{Accuracy} = \frac{\text{TP} + \text{TN}}{\text{TP} + \text{TN} + \text{FP} + \text{FN}}. \tag{27}$$

Precision refers to the ratio of the actually positive samples to the samples predicted as positive.

$$\text{Precision} = \frac{\text{TP}}{\text{TP} + \text{FP}}. \tag{28}$$

Recall refers to the ratio of the samples predicted as positive to actually positive samples.

$$\text{Recall} = \frac{\text{TP}}{\text{TP} + \text{FN}}. \tag{29}$$

F1 refers to the harmonic mean of precision and recall, which measures the robustness of the classification model.

$$F1 = \frac{2 \times \text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}. \tag{30}$$

4.2. Single Modal Hypergraph Model Performance. This section investigates the performance of a single modal hypergraph models built based on different feature extraction methods and provides references for the experiment in Sections 4.3 and 4.4 about which modals should be selected for feature fusion. Table 2 compares the classification performance of a single modal hypergraph model built based on different feature extraction methods on image dataset Cat & Dog and text dataset Ctrip.

On Cat & Dog datasets, PHA, VGG, and ResNet represent extracted image features by perceptual harsh [26], VGG [27], and ResNet [28] methods. HSIFT and HVGG represent extracting image features by SIFT [29] and VGG methods after the images are preprocessed with a color difference histogram. RVGG represents extracting image features by the VGG method after the images are Soble sharpened to enhance edge information [30]. After using the above-mentioned methods to extract features, the Euclidean distance is applied to calculate the sample similarity. SIFT represents using the key point matching number to denote the similarity between samples, after extracting the features of images' key points with SIFT.

On Ctrip datasets, TF-IDF, LSI, Word2Vec [31], and Doc2vec [32] represent using these methods to extract data features and using Euclidean distance to calculate sample similarity. Jaccard represents using the Jaccard computing method to measure the similarity between text samples. After obtaining the sample similarity, the hyperedge and hypergraph structures are built according to the KNN method.

It is known from Table 2 that, on Cat & Dog image dataset, the classification performance of the hypergraph model built based on the RVGG method is the best, and the model built based on the PHA method has the poorest quality. On the Ctrip text dataset, the model built based on the Doc2vec method is the best, and the model built based on the Jaccard method has the poorest quality. Thus, the single modal hypergraph models built with different feature extraction methods are different in classification ability.

Table 1: Test datasets.

Datasets	Datasets type	Total samples	Numbers of samples in training set	Categories
Cat & Dog	Image	2000	200	2
Cifar 10	Image	40000	6000	10
Ctrip	Text	7766	700	2
Spambase	Numerical value	4601	400	2

Table 2: Classification performance of single modal hypergraph model.

Datasets	Feature extraction method	Accuracy	Precision	Recall	F1
Cat & Dog	PHA	0.552	0.554	0.574	0.563
	SIFT	0.658	0.660	0.663	0.661
	HSIFT	0.669	0.674	0.678	0.675
	VGG	0.958	0.960	0.957	0.958
	ResNet	0.960	0.961	0.958	0.959
	HVGG	0.963	0.962	0.963	0.962
	RVGG	0.965	0.965	0.965	0.965
Ctrip	Jaccard	0.508	0.505	0.513	0.508
	TF-IDF	0.603	0.609	0.613	0.610
	LSI	0.608	0.612	0.620	0.615
	Word2vec	0.639	0.748	0.651	0.696
	Doc2vec	0.651	0.870	0.657	0.748

Therefore, it is necessary to find a suitable modal combination to perform incidence matrix extension and Laplacian matrix fusion on multimodal feature fusion hypergraphs.

4.3. Model Performance under Modal Combination. This section analyzes the impact of different model combinations, incidence matrix extension, and Laplacian matrix fusion on the hypergraph model's classification performance on text and image datasets, as shown in Table 3. In the table, "Poor + Poor/PHA + SIFT" indicates a new model obtained by combining PHA and SIFT modals. According to the results of the table, the single modal hypergraph model built by PHA and SIFT methods has the poorest quality, belonging to a "poor + poor" combination approach. For each dataset, the second combination approach is a modal combination fusing the best quality and the poorest quality, and the third combination approach is a modal combination fusing the two best qualities.

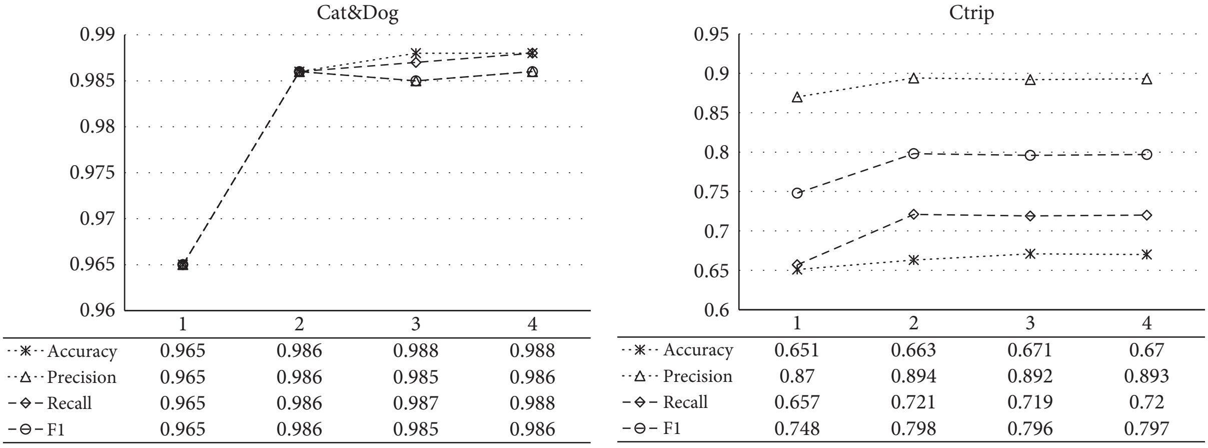
Comprehensively analyzing Tables 2 and 3, when the hypergraph model quality of two models is equivalent, the incidence matrix extension and Laplacian matrix fusion method are beneficial to improve the classification performance of the new model. On Cat & Dog dataset, Table 2 shows that the quality of the PHA or SIFT-based single modal model is the poorest, and Table 3 shows that the performance of the PHA + SIFT combination is better than PHA and SIFT separately. Likewise, the quality of RVGG and HVGG based single modal models is the best in Table 2, and the performance of the RVGG + HVGG combination in Table 3 is higher than RVGG and HVGG separately. However, when two models' model qualities are varied too much, the incidence matrix extension and Laplacian matrix fusion go on improving model performance. As shown in Table 3, the performance of the RVGG + PHA combination model is lower than RVGG, because the PHA-based single modal model quality is the poorest, which will hinder the performance of the fused new model. On Ctrip datasets, the same conclusion is made as well.

According to the data in Table 3, it is also discovered that after combining different models, the performance of models receiving double fusion of incidence matrix extension and Laplacian matrix is better than that obtained by only incidence matrix extension. Thereby, when choosing modal combinations, it is essential to choose single modal hypergraphs with higher and equivalent quality for the combination and then perform incidence matrix extension and Laplacian matrix fusion, to train and obtain high-quality models.

4.4. Influence of Modal Quantity. This section investigates the influence of fused modal quantity on hypergraph model performance, as shown in Figure 4. In the figure, the abscissa axis indicates the fused modal quantity, the data at the bottom of the table reflects the model's classification indexes obtained after fusing multiple modals, and the four curves reflect the tendency that the classification indexes increase with modal quantity. Based on the conclusion of Table 3, the single modal combination with higher and equivalent quality can obtain a higher classification performance, so 1–4 modals are selected for fusion, respectively, following a descending order of quality. Therein, on Cat & Dog datasets, the 1–4 of abscissa axes correspond to such four modal combinations as RVGG, RVGG + HVGG, RVGG + HVGG + VGG, and RVGG + HVGG + VGG + Resnet, respectively. On Ctrip datasets, the 1–4 of abscissa axes correspond to such four modal combinations as Doc2vec, Doc2vec + Word2vec, Doc2vec + Word2vec + LSI, and Doc2vec + Word2vec + LSI + TF-IDF. In Figure 4, the

Table 3: Classification performance of modal combinations.

Datasets	Modal combinations	Incidence matrix extension				Incidence matrix extension & Laplacian matrix fusion			
		Accuracy	Precision	Recall	F1	Accuracy	Precision	Recall	F1
Cat & Dog	Poor + Poor PHA + SIFT	0.687	0.662	0.674	0.667	0.722	0.718	0.720	0.718
	Poor + Good PHA + RVGG	0.919	0.920	0.919	0.919	0.928	0.926	0.930	0.928
	Good + Good RVGG + HVGG	0.975	0.975	0.977	0.976	0.986	0.986	0.986	0.986
Ctrip	Poor + Poor Jaccard + TF-IDF	0.619	0.620	0.619	0.619	0.623	0.621	0.623	0.621
	Good + Poor TF-IDF + Doc2vec	0.646	0.821	0.613	0.701	0.648	0.837	0.624	0.711
	Good + Good Doc2vec + word2vec	0.659	0.878	0.709	0.785	0.722	0.718	0.720	0.718



	1	2	3	4
Accuracy	0.965	0.986	0.988	0.988
Precision	0.965	0.986	0.985	0.986
Recall	0.965	0.986	0.987	0.988
F1	0.965	0.986	0.985	0.986

(a)

	1	2	3	4
Accuracy	0.651	0.663	0.671	0.67
Precision	0.87	0.894	0.892	0.893
Recall	0.657	0.721	0.719	0.72
F1	0.748	0.798	0.796	0.797

(b)

Figure 4: Influence of modal quantity on model performance. (a) Cat & Dog dataset. (b) Ctrip dataset.

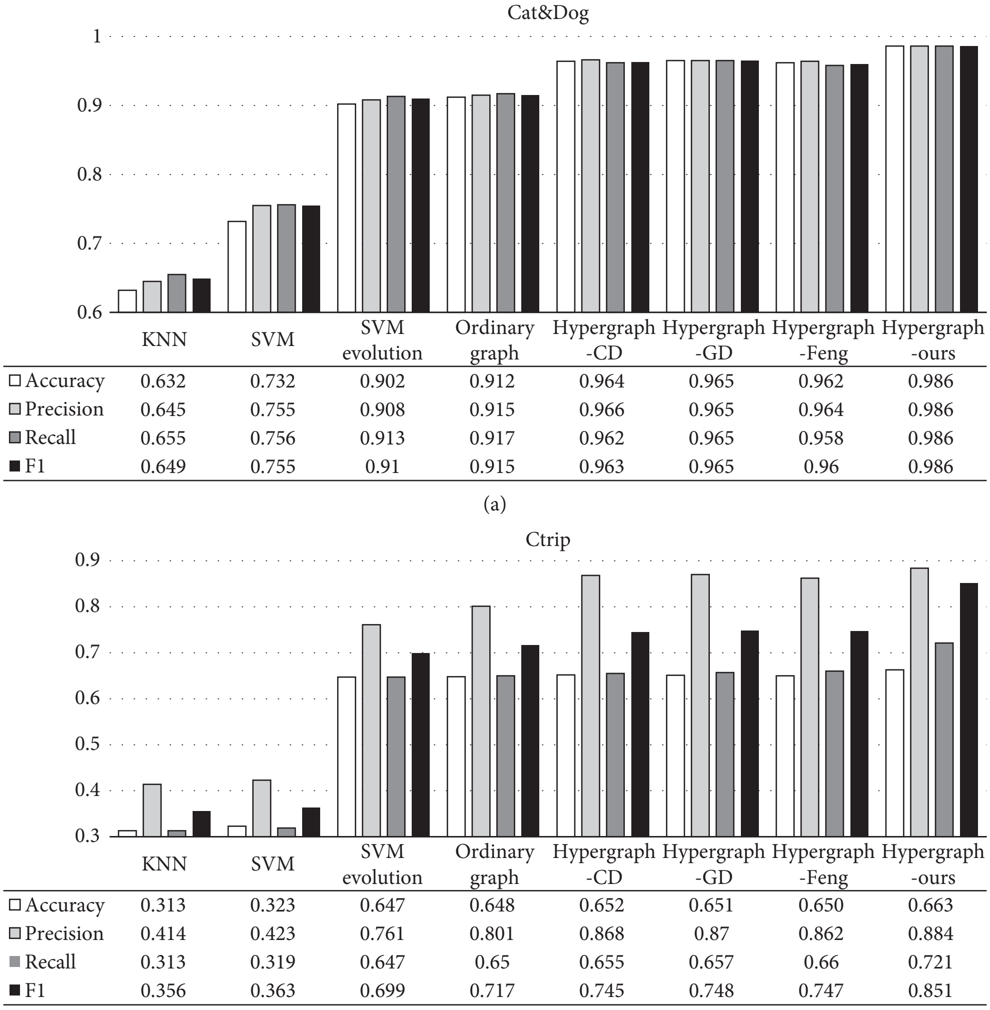
growth trend of the four curves shows that after the fused modal number is larger than 2, the model performance basically tends to be stable. Therefore, in the test of Table 4, all models only fuse two optimal modal combinations.

4.5. Time Cost. Table 4 compares the classification performance and time cost of the proposed model with the four single modal models with the best quality on four datasets. Considering Spambase datasets are numeric data and do not involve feature extraction operation, the Euclidean distance and cosine distance are used to figure out the intersample similarity, respectively, and obtain two different single modal hypergraph models, and then, the incidence matrix extension and Laplacian matrix fusion are used, respectively, to compare with the classification performance of single modal models. The experimental results in Table 4 indicate that the proposed multimodal feature fusion method can effectively promote the performance of the hypergraph learning model. Although solely using the incidence matrix extension method may result in larger cost time due to the matrix dimension problem, applying the Laplacian matrix fusion method can effectively reduce the model's time cost only slightly higher than the single modal model with the best quality. On the four datasets, the Laplacian matrix fusion can reduce the time cost by 44.2%, 44.6%, 30.3%, and 40.7% than the incidence matrix extension method, with an average reduction of 40% around.

4.6. Comparison with Other Models. This section compares the classification performance of the proposed model with other typical machine learning models, such as KNN, SVM, SVM evolutionary model, ordinary graph model [34], hypergraph model CD, hypergraph model Gd, and hypergraph model Feng. SVM evolutionary model uses the evolutionary algorithm to search the SVM variable value, while the variables of the SVM model are the default initial values of the software package. The hypergraph model CD refers to the hypergraph model obtained by solving the target function with the coordinate descending method, corresponding to equations (13)–(14). The hypergraph model Gd refers to the hypergraph model obtained by

Table 4: Time cost comparison.

Datasets	Model	Accuracy	Precision	Recall	F1	Time (s)
Cat & Dog	RVGG	0.965	0.965	0.965	0.965	20.26
	RVGG + HVGG Incidence matrix extension	0.975	0.975	0.977	0.976	49.04
	RVGG + HVGG Incidence matrix extension + Laplacian matrix fusion	0.986	0.986	0.986	0.986	27.35
Cifar 10	RVGG	0.561	0.570	0.564	0.567	11236
	RVGG + HVGG Incidence matrix extension	0.564	0.573	0.573	0.573	23793
	RVGG + HVGG Incidence matrix extension + Laplacian matrix fusion	0.594	0.613	0.583	0.598	13158
Ctrip	Doc2vec	0.651	0.870	0.657	0.748	59.7
	Doc2vec + word2vec Incidence matrix extension	0.659	0.878	0.709	0.785	106.86
	Doc2vec + word2vec Incidence matrix extension + Laplacian matrix fusion	0.663	0.884	0.721	0.851	74.2
Spambase	Euclidean	0.646	0.701	0.646	0.672	8.17
	Cosin + Euclidean incidence matrix extension	0.654	0.729	0.654	0.689	17.3
	Cosin + Euclidean incidence matrix extension + Laplacian matrix fusion	0.712	0.734	0.696	0.714	10.25



	KNN	SVM	SVM evolution	Ordinary graph	Hypergraph -CD	Hypergraph -GD	Hypergraph -Feng	Hypergraph -ours
□ Accuracy	0.632	0.732	0.902	0.912	0.964	0.965	0.962	0.986
▨ Precision	0.645	0.755	0.908	0.915	0.966	0.965	0.964	0.986
▩ Recall	0.655	0.756	0.913	0.917	0.962	0.965	0.958	0.986
■ F1	0.649	0.755	0.91	0.915	0.963	0.965	0.96	0.986

(a)

	KNN	SVM	SVM evolution	Ordinary graph	Hypergraph -CD	Hypergraph -GD	Hypergraph -Feng	Hypergraph -ours
□ Accuracy	0.313	0.323	0.647	0.648	0.652	0.651	0.650	0.663
▨ Precision	0.414	0.423	0.761	0.801	0.868	0.87	0.862	0.884
▩ Recall	0.313	0.319	0.647	0.65	0.655	0.657	0.66	0.721
■ F1	0.356	0.363	0.699	0.717	0.745	0.748	0.747	0.851

(b)

Figure 5: Comparison of model performance. (a) Cat & Dog dataset. (b) Ctrip dataset.

solving the target function with the gradient descending method, corresponding to equations (11)–(12). Hypergraph learning Feng refers to the model optimized by the incidence matrix proposed by Feng et al. [16]. For the Ctrip text dataset and Cat & Dog image dataset, it is first to extract features by Word2vec and RVGG approaches and then classify these comparison models compared with the proposed model (refer to Table 1 for the data set division of semisupervised model). 70% of the data set division of the classical classification model is a training set.

It can be seen from Figure 5 that the proposed multimodal fused hypergraph learning model has a better classification performance than typical models such as KNN, SVM, and ordinary graph. Both hypergraph model CD and hypergraph model GD models are single modal models built based on a single feature extraction method, with inferior performance than the proposed model. Although the hypergraph model Feng optimizes the incidence matrix, it ignores the information of hyperedge weight, so its effect is poorer than the proposed model. From the comparison in Figure 5, the proposed multimodal feature fusion method, by virtue of matrix extension and Laplacian matrix fusion, breaks the limits of the single hypergraph structure, fuses multimodal abstract features, and promotes the performance of the hypergraph model.

5. Conclusion

Existing hypergraph learning models are all single modal models built based on one feature extraction method, with limited feature extraction and abstract expression ability. This paper proposed a multimodal feature fusion method, making the single modal hypergraph structures built based on different feature extraction methods to fuse multimodal abstract features through extending the incidence matrix and its weight matrix. Then, by using the Laplacian matrix fusion method, every modal's incidence matrix and weight matrix receive Laplacian matrix transformation, respectively, and then undertake weighted accumulation for further model training. In this way, not only the model's time cost is reduced but also the model performance is further improved. As the hypergraph neural model is put forward, hypergraph shows its efficient learning ability in the deep learning field. Therefore, the future study will focus on the hypergraph neural network model and its optimization.

Data Availability

The data underlying the results presented in the study are available within the article.

Conflicts of Interest

The authors declare that they have no conflicts of interest.

Acknowledgments

The authors acknowledge the National Natural Science Foundation of China (Grant nos. 61772356 and 61876117), Six Talent Peak Project of Jiangsu Province (XYDXX-084), Hong Kong Research Grant Council (GRF 11211519), CERNET Innovation Project (NGII20190314), China Postdoctoral Science Foundation (2020M671597), Jiangsu Postdoctoral Research Foundation (2020Z100), Suzhou Planning Project of Science and Technology (Grant nos. SNG2020073 and SS202023). And this work was also supported by Project Funded by the Priority Academic Program Development of Jiangsu Higher Education Institutions.

References

[1] Z. Wu, S. Pan, F. Chen, and G. C. P. S. Long, "A comprehensive survey on graph neural networks," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 32, no. 1, pp. 4–24, 2021.

[2] H. Tang, G. Zhao, X. Bu, and X. Qian, "Dynamic evolution of multi-graph based collaborative filtering for recommendation systems," *Knowledge-Based Systems*, vol. 228, Article ID 107251, 2021.

[3] D. Zhou, J. Huang, and B. Schölkopf, "Learning with hypergraphs: clustering, classification, and embedding," in *Proceedings of the 19th International Conference on Neural Information Processing Systems*, pp. 1601–1608, ACM, Vancouver, CA, USA, 2007.

[4] I. R. M. Association and Y. Huang, "Hypergraph based visual segmentation and retrieval," *Graph-Based Methods in Computer Vision: Developments and Applications*, IGI Global, vol. 1, , pp. 118–139, 2013.

[5] M. Liu, J. Zhang, P.-T. Yap, and D. Shen, "View-aligned hypergraph learning for Alzheimer's disease diagnosis with incomplete multi-modality data," *Medical Image Analysis*, vol. 36, pp. 123–134, 2017.

[6] Z. Chen, G. Yue, M. Brent et al., "Identifying disease-related subnetwork connectome biomarkers by sparse hypergraph learning," *Brain Imaging and Behavior*, vol. 13, no. 4, pp. 879–892, 2018.

[7] Y. Guo, P. Dong, and S. Hao, "Automatic segmentation of Hippocampus for longitudinal infant brain MR image sequence by spatial-temporal hypergraph learning," *Patch-based techniques in medical imaging*, vol. 9993, pp. 1–8, 2016.

[8] C. Yang, R. Wang, S. Yao, and T. Abdelzaher, "Hypergraph learning with line expansion," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 669–678, IEEE, Orlando, Fl, USA, May 2020.

[9] Z. Chen, G. Yue, M. Brent et al., "Identifying high order brain connectome biomarkers via learning on hypergraph," in *Proceedings of the International Workshop on Machine Learning in Medical Imaging*, pp. 1–9, MLML, Athens, Greece, October 2016.

[10] F. Luo, L. Zhang, X. Zhou, T. Guo, Y. Cheng, and T. Yin, "Sparse-adaptive hypergraph discriminant analysis for hyperspectral image classification," *IEEE Geoscience and Remote Sensing Letters*, vol. 17, no. 6, pp. 1082–1086, 2020.

[11] Y. Gao, W. Meng, H. B. Luan, J. Shen, S. Yan, and D. Tao, "Tag-based social image search with visual-text joint hypergraph learning," in *Proceedings of the 19th International Conference on Multimedia*, pp. 1517–1520, ACM, Scottsdale, AZ, USA, November 2011.

[12] S. Bandyopadhyay, K. Das, and M. N. Murty, "Hypergraph attention isomorphism network by learning line graph expansion," in *Proceedings of the 2020 IEEE International Conference on Big Data (Big Data)*, pp. 669–678, IEEE, Atlanta, GA, USA, December 2020.

[13] T. Hu and M. K, “Multi-terimnal flows in hypergraphs,” *VLSI Circuit Layout: Theory and Design*, pp. 87–93, 1985.

[14] S. Agarwal, K. Branson, and S. Belongie, “Higher order learning with graphs. Higher order learning with graphs,” in *Proceedings of the ACM international conference on Machine learning*, pp. 17–24, ACM, Pennsylvania, USA, June 2006.

[15] G. Wang, K. Wang, H. Wang, H. Lu, X. Zhou, and Y Feng, “Uncovering local community structure on line graph through degree centrality and expansion,” *International Journal of Modern Physics B*, vol. 35, no. 08, Article ID 2150120, 2021.

[16] Y. Feng, H. You, Z. Zhang, R. Ji, and Y. Gao, “Hypergraph neural networks,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, pp. 3558–3565, AAAI Press, Hawaii, USA, September 2019.

[17] Y. Dong, W. Sawin, and Y. Bengio, “HNHN: hypergraph networks with hyperedge neurons,” 2020, https://arxiv.org/abs/2006.12278.

[18] P. Guo, R. F. Li, and H. Hu, “A clustering learning method based on hypergraph Markov chain slack,” *Computer Science*, vol. 46, no. B06, 5 pages, 2019.

[19] Z. Zhang, Y. Liang, and E. Hancock, “Joint hypergraph learning and sparse regression for feature selection,” *Pattern Recognition*, vol. 63, pp. 291–309, 2017.

[20] K. Pliakos and C. Kotropoulos, “Personalized and geo-referenced image recommendation using unified hypergraph learning and group sparsity optimization,” in *Proceedings of the International Symposium on Communications*, pp. 306–309, IEEE, Athens, Greece, May 2014.

[21] L. Nong, J. Wang, J. Lin, H. Qiu, L. Zheng, and W. Zhang, “Hypergraph wavelet neural networks for 3D object classification,” *Neurocomputing*, vol. 463, pp. 580–595, 2021.

[22] Dogs & Cats Images[Db/Ol]: https://www.kaggle.com/.

[23] Cifar-10 dataset [Db/Ol]: https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz.

[24] Tan Songbo Corpus[Db/Ol]: http://lcc.software.ict.ac.cn/-tansongbo/corpusl.php.

[25] Machine Learning Repository[Db/Ol]: http://archive.ics.uci.edu/ml.

[26] H. X. Liu, X. J. Wu, and J. Yu, “Trans-modal search algorithm of combined harsh feature and classifier learning,” *Pattern Recognition and Artificial Intelligence*, vol. 33, no. 2, p. 6, 2020.

[27] K. Simonyan and A. Zisserman, “Very deep convolutional networks for large-scale image recognition,” *Computer Science*, vol. 12, 2014.

[28] Z. Allen-Zhu and Y. Li, “What can resnet learn efficiently, going beyond kernels?” *Advances in Neural Information Processing Systems*, vol. 32, 2019.

[29] Y. Wang, Y. Yuan, and Z. Lei, “Fast SIFT feature matching algorithm based on geometric transformation,” *IEEE Access*, vol. 8, no. 99, pp. 88133–88140, 2020.

[30] X. Y. Luo, F. L. Liu, C. F. Yang, and C. Yang, “PS image and steganographic image detection based on noise model and feature combination,” *Chinese Journal of Computers*, vol. 8, no. 6, p. 13, 2010.

[31] G. Di Gennaro, A. Buonanno, and F. A. N. Palmieri, “Considerations about learning Word2Vec,” *The Journal of Supercomputing*, vol. 77, no. 11, pp. 12320–12335, 2021.

[32] S. Kim, I. Park, and B. Yoon, “SAO2Vec: d,” *PLoS One*, vol. 15, no. 2, Article ID e0227930, 2020.