KNH: Multi-View Modeling with K-Nearest Hyperplanes Graph for Misinformation Detection

Sara Abdali
sabda005@ucr.edu
University of California, Riverside

Neil Shah
nshah@snap.com
Snap Inc.

Evangelos E. Papalexakis
epapalex@cs.ucr.edu
University of California, Riverside

ABSTRACT

Graphs are one of the most efficacious structures for representing datapoints and their relations, and they have been largely exploited for different applications. Previously, the higher-order relations between the nodes have been modeled by a generalization of graphs known as hypergraphs. In hypergraphs, the edges are defined by a set of nodes i.e., hyperedges to demonstrate the higher order relationships between the data. However, there is no explicit higher-order generalization for nodes themselves. In this work, we introduce a novel generalization of graphs i.e., K-Nearest Hyperplanes graph (KNH) where the nodes are defined by higher order Euclidean subspaces for multi-view modeling of the nodes. In fact, in KNH, nodes are hyperplanes or more precisely m-flats instead of datapoints. We experimentally evaluate the KNH graph on two multi-aspect datasets for misinformation detection. The experimental results suggest that multi-view modeling of articles using KNH graph outperforms the classic KNN graph in terms of classification performance.

KEYWORDS

K-Nearest Hyperplanes Graph, Multi-View Modeling, Fake News Detection, Tensor Decomposition, Canonical Correlation Analysis

1 INTRODUCTION

The evolution of data storage technologies has made data scientists capable of storing huge volume of information and analyzing the data considering hundreds of aspects or features. Although accessing more information brings about a more holistic view of data points, the classification task for labeling the datapoints considering different aspects of the data has become a more challenging task. To this end, variety of techniques under the umbrella of ensemble learning approaches have been introduced by machine learning researchers. The ensemble learning approaches aim at combining individual classifiers often designed for one aspect of the data, to create a robuster classifier that merges the decision making process of all individual ones[6, 30].

Over the last decades, multiple approaches have been introduced for data representation. Graphs are one of the most efficacious data structures employed extensively by mathematicians and computer scientists for countless applications among which we can mention fake news detection and article classification. For instance, in [1, 3], the graph data structure in form of a K-Nearest Neighbours graph (KNN) is used to model similarity of the articles (nodes) and the pairwise relationship as edges that connect them. Unfortunately, the traditional KNN graph is not explicitly capable enough for multi-view representation of entities (nodes) which is often one of the requirements of ensemble learning. However, there are works that merge all aspects into a joint structure to use it for construction

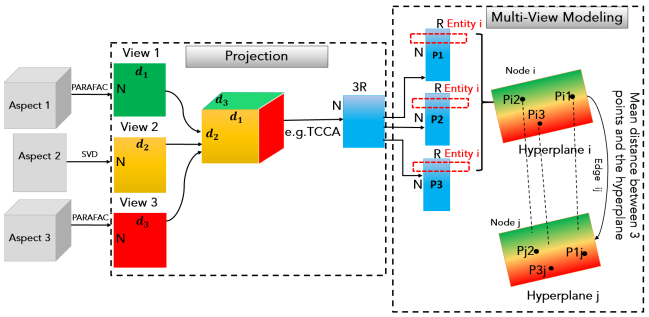


Figure 1: An overview of proposed K-Nearest Hyperplane graph (KNH) for multi-view modeling and ensemble learning.

of the graph. For instance, to take advantage of multi-aspect information and KNN graph at the same time, in [1] joint patterns are derived from different aspects of the news articles and then the patterns are leveraged to construct a KNN graph and classify the unknown articles. Another way that comes into mind in order to use KNNs for higher order representation is to create multiple of graphs for different aspects and then merge them somehow but considering the high dimensionality of real world datasets, this solution is not only very expensive but also makes the merging step of ensemble learning very complicated. In other words, finding an insightful way to combine all aspects of the nodes seems almost impractical.

However, an extension of graph models known as hypergraph has been introduced where an edge may connect more than two nodes to illustrate the higher order relationships between the nodes. In fact, instead of a single weighted connection, an edge is a subset of nodes that are similar in terms of features or distance [7, 31]. There are previous works that leverage hypergraphs for variety of machine learning tasks. For example, in [29], for image classification task, Li. et al. propose an adaptive hypergraph learning method that varies the size of the neighborhood and generates multiple hyperedges for each sample. Same as previous one, Yu. et al. also leverage hypergraph data structure for image classification task. In [25], Lian. et al. present a hypergraph based formulation for multi-label classification task. In this work, the hypergraph is created to use the correlation information among different labels. There are aslo some previous works that leverage hypergraphs for object detection task [18].

Having said that, the main focus of hypergraphs is on hyperedges or high order relationships between the nodes and they do not explicitly represent higher order representation of the node itself.

In this work, we propose a novel generalization of graphs applicable for news article classification, even though it could be leveraged

for any multi-view representation task. The intuition behind this work is to define a common "feature space" comprising multiple-views of nodes i.e., articles in this work, using geometric objects which is not only capable of multi-view modeling of the articles but also could be exploited to *predict missing or arriving features. Moreover, defining a common feature space to model articles enables us to use geometric techniques to calculate intersection, orthogonality, linearity, distance etc. between different articles (nodes).* Contrary to the hypergraph learning techniques which aim at defining hyperedges and their weights to model high order relationships between nodes, the goal of this work is to model higher order representation of the nodes (articles).

To this end, we propose to first capture the *entity (article) views with respect to different aspects* and then map these views to a new shared space so that we can use the mapped views for defining higher order geometric objects which produces a manifold representation of the articles.

The contribution of this work are as follows:

- **A novel graph based modeling for multi-view representation of articles using Euclidean subspaces** We introduce a generalization of KNN graphs where instead of datapoint as nodes in N-dimensional space, there are hypernodes defined by M-flats $M < N$ (subspaces in N-dimensional space). Each hypernode (subspace) is defined by multiple datapoints (views) for each entity (article) and present a manifold representation of it.
- **A novel decomposition-based pipeline for ensemble learning** We introduce a novel decomposition-based pipeline that leverages KNH graphs for ensemble learning and multi-view classification of articles. This pipeline consists of decomposition (CP or SVD), tensor canonical analysis (TCCA), graph modeling and edge defining.
- **Experimenting on real world datasets** We examine the KNH modeling and classification pipeline on two real world datasets including textual, user and social context aspects of news articles.

The organization of the paper is as follows:

We first present the related work in section 2.1 and then we discuss the mathematical background required for KNH modeling and the classification pipeline. Next, we state the problem formulation and then in section 3 we describe the proposed KNH method. In Section 6, we examine the proposed method on two real world datasets and finally we conclude.

2 RELATED WORK

2.1 Ensemble Learning for Fake News Detection

The majority of misinformation detection approaches focus on a single aspect of the data and mostly the article content [14, 27]. There are also works that leverage other aspects like user features [28], and temporal properties [17]. However, there exist few ensemble approaches that consider all different aspects simultaneously. For instance, in [12] the authors propose an ensemble model by merging a bag of words embedding, user-user, user-article and publisher-article interactions. In another work [13], news contents and user comments are consolidated to detect the misinformation jointly. Another example is [1], where content-based, social-context in form of hashtags and website features are leveraged to create manifold patterns for multi-aspect detection of misinformation. In this work, we leverage the promising aspects introduced in both [12] and [1] but this time with a different and novel multi-aspect modeling and formulation.

2.2 Hypergraph Learning

The hypergraphs are one extension of graph models in which an edge can connect more than two nodes. In other words, an edge is defined as a subset of nodes[7, 31] that share same (similar) feature. In contrast to traditional graph-based learning methods which only model the pairwise relationship between entities, the hypergraph leverage hyperedges to model higher-order relationships between the entities. In previous work, the hypergraph learning has been used for variety of machine learning applications. For instance, in [29] Li. et al. propose to model an image as a hypergraph that leverages hyperedges to capture the contextual features of the pixels. In another work, Lian. et al. construct a hypergraph to exploit the correlation information among labels for multi label classification task[25]. In [18] Yu. et al. propose an adaptive hypergraph based method for classification of images. Moreover, there are previous works that leverage hypergraphs for object detection tasks[19, 24]. In hypergraphs, the main focus is to define hyperedges and the weights to model high order relationships. Although there are some unsupervised work using affinities within the hyperedges, [11], hypergraphs, do not have exploratory capabilities to define a common "feature space" to predict behaviour of arriving data points (nodes) or predicting the missing data points using this common space. Moreover, finding the weights for the hyperedges is a challenging task and requires complicated optimization and regularization techniques like graph Laplacian, $\mathbf{L}_1$ and $\mathbf{L}_2$ regularizers [18, 26]. In this work, we try to present a generalization of learning graphs that mostly focuses on defining hypernodes or a common "feature space" for multiple-views of entities in dataset which not only is capable of multi-view modeling of the data but also can be exploited to predict missing or arriving data.

3 BACKGROUND

In this section, we first present mathematical background requires for the proposed method and then we discuss the problem definition and proposed K-Nearest Hyperplanes Graph (KNH).

3.1 Matrix and Tensor Decompositions

A tensor is an array with three or more than three dimensions where the dimensions are usually referred to as modes[22, 23]. In linear algebra, there is a factorization algorithm known as Singular Value Decomposition (SVD) in which we can factorize a matrix $\boldsymbol{X}$ into the product of three matrices as follows:

$$\boldsymbol{X} \approx \mathbf{U}\Sigma\mathbf{V}^T \tag{1}$$

where the columns of U and V are orthonormal and the matrix Σ is a diagonal with positive real entries. Using rank $\mathbf{R}$, SVD decomposition we can represent a matrix as a summation of $\mathbf{R}$ rank 1 matrices as follows:

$$\boldsymbol{X} \approx \Sigma_{r=1}^{R} \sigma_r \mathbf{u}_r \circ \mathbf{v}_r \tag{2}$$

Table of Notations

Symbol	Definition
$\mathcal{X}$,$\mathbf{X}$,x	Tensor,Matrix,vector
$\circ$	Outer product
$\times$	Cross product
$Cov(x, y)$	Covariance x and y
$E(x)$	Mean x
$\rho = Corr(x, y)$	Correlation between x&y
C_{xx}	Variance matrix of vector x
C_{xy}	Covariance matrix of vectors x&y
$C_{12\cdots m}$	Covariance Tensor
h_x	Canonical vector
z_x	Canonical variable

Table 1: Symbols and Definitions

The Canonical Polyadic (CP) or PARAFAC decomposition is an extension of SVD for higher mode matrices i.e., tensors [9]. Indeed, CP/PARAFAC factorizes a tensor into a summation of rank-one tensors. For instance, a three-mode tensor is decomposed into a sum of outer products of three vectors as follows:

$$\mathcal{X} \approx \Sigma_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{3}$$

where $\mathbf{a}_r \in \mathbb{R}^I$, $\mathbf{b}_r \in \mathbb{R}^J$, $\mathbf{c}_r \in \mathbb{R}^K$ and the outer product is given by [22, 23]:

$$(\mathbf{a}_r, \mathbf{b}_r, \mathbf{c}_r)(i, j, k) = \mathbf{a}_r(i)\ \mathbf{b}_r(j)\ \mathbf{c}_r(k)\ \forall i, j, k \tag{4}$$

Factor matrices are defined as $\mathbf{A} = [\mathbf{a}_1\ \mathbf{a}_2 \dots \mathbf{a}_R]$, $\mathbf{B} = [\mathbf{b}_1\ \mathbf{b}_2 \dots \mathbf{b}_R]$, and $\mathbf{C} = [\mathbf{c}_1\ \mathbf{c}_2 \dots \mathbf{c}_R]$ where R is the rank of decomposition or the number of columns in the factor matrices. The optimization problem for finding factor matrices is as follows:

$$\min_{A,B,C} = \|\mathcal{X} - \Sigma_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r\|^2 \tag{5}$$

One effective way for solving the optimization problem above is to use Alternating Least Squares (ALS) which solves for any of the factor matrices by fixing the others [22, 23].

3.2 Canonical Correlation Analysis (CCA)

In 2-dimensional space the correlation between two vectors x, y is defined as follows [20, 21]:

$$\begin{aligned} \rho &= Corr(x, y) \\ &= \frac{cov(x, y)}{\sigma_x \sigma_y} \end{aligned} \tag{6}$$

Since $Cov(x, y) = E(xy) - E(x)E(y) = E(xy)$ [5], if we suppose the vectors are centered around the mean, then $E(x)$ and $E(y)$ are zero and ρ is going to be[21]:

$$\rho = \frac{E(xy)}{\sqrt{E(x^2)E(y^2)}} \tag{7}$$

There is a technique known as Canonical Correlation Analysis or CCA which we can use to find canonical vectors h_x, h_y such that if we project two vectors x and y using these two canonical vectors into canonical variables z_x, z_y, the correlation between z_x and z_y is maximized [20, 21]:

$$\begin{aligned} argmax\rho_{z_1,z_2} &= corr(z_1, z_2) \\ &= \frac{E(h_x^T x y^T h_y)}{\sqrt{E(h_x^T x x^T h_x)E(h_y^T y y^T h_y)}} \\ &= \frac{h_x^T C_{xy} h_y}{\sqrt{h_x^T C_{xx}^T h_x h_y^T C_{yy} h_y}} \end{aligned} \tag{8}$$

Where $C_{xx} = XX^T$,$C_{yy} = YY^T$ are variance matrices and $C_{xy} = XY^T$ is covariance matrix of vectors x and y.

3.2.1 Tensor Canonical Correlation Analysis (TCCA). When we have more than two variables, we can also define the optimization problem above as a minimization problem where we aim at minimizing the pairwise distance between the variables. So, the generalized form of the CCA can be redefined as follows [21]:

$$argmin_{h_p} \frac{1}{2m(m-1)} \Sigma_{p,q=1}^{m} \|X_p^T h_p - X_q^T h_q\|^2 \tag{9}$$

As we know $C_{xy} = XY^T$ is equal to covariance matrix of x and y. In higher dimensional space we can also define variance matrix C_{pp} and covariance tensor $C_{1\cdots m}$ as follows [21]:

$$C_{12\cdots m} = \frac{1}{M} \Sigma_{n=1}^{m} x_{1n} \circ x_{2n} \circ \cdots \circ x_{mn} \tag{10}$$

$$C_{pp} = \frac{1}{M} \Sigma_{n=1}^{m} x_{pn} x_{pn}^T \quad p \in 1 \dots m \tag{11}$$

We can show that higher order canonical correlation can be computed by CP/ALS optimization probelm. For proof you can refer to [21].

3.3 Hyperplanes and flats in n-dimensional space

A hyperplane in an n-dimensional space V is an $n-1$ dimensional subspace which is defined by following linear equation [4]:

$$a_1(x_1 - x_1{}') + a_2(x_2 - x_2{}') + \cdots + a_n(x_n - x_n{}') = 0 \tag{12}$$

Where the vector $(a_1, a_2, \dots, a_n)$ is a normal vector perpendicular to the hyperplane and $(x_1{}', x_2{}', \dots, x_n{}')$ is a point on the hyperplane. Therefore, we can rewrite the linear equation of hyperplane as[4]:

$$a_1 x_1 + a_2 x_2 + \cdots + a_n x_n = d \tag{13}$$

Given n datapoints we can uniquely define a hyperplane in an n-dimensional space. The distance from a point $(x_1{}', x_2{}', \dots, x_n{}')$ to a hyperplane is defined as follows[4]:

$$d_{point-hyperplane} = \frac{|a_1 x_1{}' + a_2 x_2{}' + \cdots + a_n x_n{}' + d|}{\sqrt{a_1{}^2 + a_2{}^2 + \cdots + a_n{}^2}} \tag{14}$$

A flat or Euclidean subspace is any lower dimension subspace in that space. For instance, flats in 4-dimensional space are points, lines, and planes. We can described a flat in n-dimensional space by a system of linear parametric equations. For example, the equation of a line in n-dimensional space is equal to:

$$x_1 = a_1 t + b_1, x_2 = a_2 t + b_2, \dots, x_n = a_n t + b_n \tag{15}$$

Then we can use the Euclidean distance to calculate the distance from a point to a 2-flat (line). For instance, we can calculate the

distance from point P_0 to a 2-flat (line) defined by 2 points P_1 and P_2 in 3-dimensional space as:

$$d_{point-line} = \frac{|(p_2 - p_0) \times (p_1 - p_0)|}{|p_2 - p_1|} \tag{16}$$

Where $\times$ is the cross product of two vectors $(p_1 - p_0)$ and $(p_2 - p_0)$.

Just like the previous one, we can define a 3-flat (plane) in n-dimensional space as follows:

$$x_1 = a_1 t_1 + b_1 t_2 + c_1, x_2 = a_2 t_1 + b_2 t_2 + c_2, \ldots, x_n = a_n t_1 + b_n t_2 + c_n \tag{17}$$

3.4 K-Nearest Neighbors graph (KNN)

We can model entities in a dataset using a k-nearest-neighbor graph in which each entity is a node or a datapoint in feature space and the edges between the datapoints represent the distances or similarities between the entities. [8].

What if there are multiple of views for each entity in the dataset, each of which represents the entity with respect to a specific aspect of it? The idea of this work is to find a holistic representation that comprises all different views of the entity and finding a way for calculating the distances between these manifold representation of each entity. In next section we define a novel approach for generalizing the KNN graphs and a new way for measuring the similarity between the entities.

4 PROBLEM FORMULATION

The problem formulation of multi view modeling and classification using K-Nearest Hyperplanes graph is as follows:

> **Given** a dataset comprising N entities and M matrices of size $N \times d_m$,$m = 1, \ldots, M$ for M views of the data such that row i of matrix M corresponds to a d_m-dimensional representation of entity i with respect to view m.
> **Find** a representation for the entities
> **Such that** the manifold structures are preserved when used for modeling and classification of those entities.

One simple solution that comes into mind is to stack views into a long vector and use KNN graph for modeling and classification. But by doing so, we may destroy potentially useful structures. We address this problem by defining a M dimensional flat in R dimensional space for each entity where R is the dimensionality of view matrices after projecting into a common space. For example, if we have 2 view matrices we can model each entity by a line and if we have 3 views, we model entities with a plane. These flats are generalized form of points (nodes) in KNN graph. We can then leverage geometrical properties of hyperplanes to calculate a manifold distance between entities which can be shown as graph edges. As we will see in upcoming experiments, retaining the proposed representation results in better quality in downstream classification tasks. The details are described in next section.

5 HYPERPLANE MODELING AND K-NEAREST HYPERPLANES GRAPH

In what follows the hyperplane modeling and classification will be described step by step.

5.1 Modeling the Aspects using Tensor/Matrix and Decomposing Aspects into View Matrices

Matrices and tensors are common tools for modeling entities in feature space. For instance, using the well known bag of word matrix we model documents in word space. Likewise, for multi aspect modeling of the entities, we leverage tensors such that one mode of the tensor correspond to entities and other modes represent different aspects that the entities are defined by. To capture the hidden patterns of the entities with respect to the considered aspect(s), we decompose the matrix(tensor) into factor matrices as described in previous section. Having this in mind, the very first step of the proposed approach is to decompose M models of the entities (matrices or tensors) into M entity mode factor matrices henceforth referred to as view matrices each of which of size $N \times d_m$ where N is the number of entities and d_m is the size of latent pattern space defined by rank of decomposition. In fact, each view comprises latent patterns of the entities with respect to the considered aspect.

5.2 Projecting Views into a Common Space

Previous step provides us with M pattern matrices of size $N \times d_m$, $m = 1, \ldots, M$ for M different views of the data. Now, we want to leverage all these view matrices to create a manifold description of entities. In fact, the goal is to define a new space that consolidates all M representation of the entities. Since these matrices represent the entities in different spaces, we need to find a way to project all theses different representations into a common space such that the correlation between all representations is maximized. One solution that comes into mind for this requirement is the Canonical Correlation Analysis or CCA as discussed earlier. Likewise, if we have more than 2 vectors for each entities corresponding to more than 2 view matrices, we can leverage TCCA or higher order CCA to maximize the correlation between the rows of M views. To this end, we first create a tensor $\mathcal{X}$ of size $d_1 \times d_2 \ldots \times d_m$ out of all M matrices which is equivalent to the covariance tensor. Then we leverage TCCA algorithm as explained earlier, to project all views into a new space. The rank of decomposition is equal to the dimension of the new space. As an example, suppose we have three view matrices. We define a 3-mode covariance tensor as follows:

$$\mathcal{C}_{123} \approx \Sigma_{r=1}^{R} \mathbf{m}_{1r} \circ \mathbf{m}_{2r} \circ \mathbf{m}_{3r} \tag{18}$$

Where $\mathcal{C}_{123}$ is the covariance tensor and m_1 to m_3 are view matrices and R is the rank of decomposition for finding the maximally correlated variables. Now, we leverage TCCA algorithm to solve the equation 9. It is worth mentioning that, projection when maximizing the correlation is a part and parcel of the consolidation process without which defining an integrated representation does not make sense.

5.3 Creating M-Dimensional Flat (Hyperplane) in R-Dimensional Space for Each Entity

The output of the projection step is M matrices of size $N \times R$ where the rows of each matrix corresponds to one point in R-dimensional space such that the correlation between rows i of all matrices is maximized. Now, there are M datapoints in R-dimensional space for each entity. We can leverage these points to define a flat for

each entity. For instance, if we have 2 views and the views are projected into a R-dimensional space, we can define 2-flats (lines) in R-dimensional space as follows:

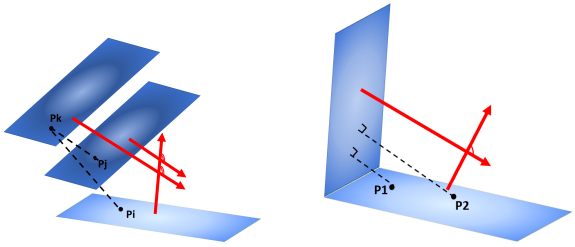
$$x_1 = a_1 t + b_1, x_2 = a_2 t + b_2, \ldots, x_n = a_R t + b_R \tag{19}$$

where the number of parametric equation is equal to $R - 2$. In general, a flat of dimension $R - k$. is described by k parametric equations.

5.4 Creating K-Nearest Hyperplane Graph for Classification

Previous step results in N, M-flats in R-dimensional space, each of which a manifold representation of entity i, $i = 1, \ldots, N$. We can create a generalized K-NN graph in such a way that each node of the graph is a M-flat in R-dimensional space and the edges between the nodes show the multilateral similarity between the flats (nodes). The question that raises here is: "how to calculate the distances between the hyperplanes?" because if we are to use the Euclidean distance between the hyperplanes, they should be parallel, otherwise the distance between them is equal to zero.

One way that comes into mind is to calculate the angle between the hyperplanes which is equal to the angle between the normal vectors of the hyperplanes, but lets consider the situation demonstrated in Figure. 2 part a, where plane j and k are parallel to plane i so, they form the same angle with plane i. In this situation, there might be a point P_i lying on plane i which is closer to a point P_j on plane j than a point P_k on plane k. Thus, the angle scenario is not capable to capture this difference. But if the point-hyperplane distance in 14 is considered, it is possible to capture an insightful difference illustrated in 2 part b. The closer the points are to the *intersection of the hyperplanes*, the smaller the $d_{point-hyperplane}$ gets.



(a) Similarity based on angles between normal vectors. As depicted, angle between the hyperplanes i.e., angle between the normal vectors is not a proper metric to measure the similarity e.g., P_i is closer to P_j than P_k, but the angle both plane form with plane i is equal.

(b) Similarity based on point-plane Euclidean distance of points of one plane to another plane. The closer the points are to the intersection of the hyperplanes, the smaller the distance gets.

Figure 2: Comparing different approaches for measuring similarity between hyperplanes.

.

Having justification above in mind, we define the following distance as the distances between the hyperplanes in KNH graph. we use the mean of Euclidean distances between each of M datapoints defining hyperplane i and hyperplane j as mentioned in equation 14. For instance, in R-dimensional space we can define the weight d_{ij} of the edges using Algorithm. 1.

In Figure 1 and algorithm 1 an overview of the KNH approach is demonstrated.

Algorithm 1: K-Nearest Hyperplanes modeling using 2-views in R-dimensional space

Input:2 embedding of size $N \times d_m$
Result: a K-nearest hyperplane graph
// Projecting all embeddings into a common R-dimensional space where w is a $N \times 2R$ matrix
$W = CCA(M_1, M_2, R)$
$P1 = w(:, 1 : R)$
$P2 = w(:, R + 1 : 2R)$
// Defining the Line
$x_1 = a_1 t + b_1, x_2 = a_2 t + b_2, \ldots, x_n = a_R t + b_R$
for *all* $i, i = 1, \ldots, N$ **do**
  for *all* $j, j = i, \ldots, N$ **do**
    $p_{1j} = P_1(j, :) - P_1(i, :)$
    $p_{2j} = P_2(j, :) - P_1(i, :)$
    $p_{12i} = P_2(i, :) - P_1(i, :)$
    $t_1 = dot(p_{1j}, p_{12i})/dot(p_{12i}, p_{12i})$
    $t_2 = dot(p_{2j}, p_{12i})/dot(p_{12i}, p_{12i})$
    $d_1 = (p_{1j} - t_1 * p_{12i})$
    $d_2 = (p_{2j} - t_2 * p_{12i})$
    $d_1 = sqrt(sum(d_1.^2))$
    $d_2 = sqrt(sum(d_2.^2))$
    $d(i, j) = (d_1 + d_2)/2$
  end
end
Create_Graph(d)

5.5 Complexity Analysis

The time complexity of KNH method depends on the time complexity of the TCCA and the construction of the graphs which consists of construction of the nodes and calculating the weight of the edges. As discussed in [21], the time complexity of TCCA is independent of the number of instances and can be scaled for large size problems and the space and time complexity of the approach are $O(N^m)$ and $O(trN^m)$ respectively[21]. To define the nodes, we need to calculate the normal vectors which is equivalent to calculating N cross product each of which of size M or number of views $O(NM)$. The complexity of calculating the edges is same as the time complexity of KNN classification and is $O(N^2)$.

6 EXPERIMENTS

In this section, we empirically evaluate the effectiveness of the proposed KNH method against traditional KNN for multi-view modeling and classification task. We experiment on a 2-aspects document-publisher dataset extracted from Twitter's tweets [1] and another 2-aspects news article dataset extracted from FakeNewsNet

[1] https://github.com/Saraabdali/Fake-News-Detection-_ASONAM-2018

dataset[2] but this time we experiment on different sets of features, i.e., user-news interaction aspects and the publisher-news interaction aspect. Henceforth, we refer to the first dataset as Twitter dataset and to the second dataset as politifact dataset. We first, introduce the details of each dataset and the extracted aspects and then we present the experimental results.

6.1 Implementation

We implemented both experiments described above in Matlab using Tensor Toolbox version 2.6 [2]. For rank of decomposition (dimensionality) r_m of each view and the number of nearest neighbors K we grid searched the values between range 1-50 for r_m and $1-30$ for K. Later on, we will show the classification trends for different ranks and number of neighbors. We measured the effectiveness of all methods using average precision, recall, F1 score and accuracy for 10 runs of each method.

6.2 Experiment 1: Article Classification using 2-Flats (Lines) Created by Textual Content and Domain Aspects

6.2.1 Description of dataset and aspects. As mentioned earlier, for the first experiment, we use the dataset introduced in [1, 3]. This dataset comprises multi-aspect information about news articles and the Twitter tweets shared these articles as URL links. In this dataset, the labels are extracted using the BSDetector Google Chrome extension [3] which is a *crowd-sourced* toolbox. In aformentioned works, the bias, clickbait, conspiracy, fake, hate, junk science, rumor, satire, and unreliable categories as considered as misinformative articles. In this work, we also follow the same strategy. Moreover, to prevent the domain bias discussed in [1], as suggested, we select one article per domain. Thus, we created a relatively balanced sample by randomly selecting one articles per domain as described in Table. 2.

Twitter dataset	
Features	**Total Number**
words	18853
Domains	652
Article	335 (Real)/317 (Fake)

Table 2: Twitter dataset description

As the base case i.e., 2-view classification of articles which corresponds to 2-flat (line) modeling, we leverage the most promising aspect models i.e., TTA and Tags introduced in [1]. The description of the models is as follows:

- **(Term, Term, Article) Tensor**: As suggested in [3, 10] different classes of news articles, i.e., misinformative and real classes tend to have some common words that co-occur within the text. The co-occurrence of the words forms some patterns which is shared between different categories of the articles. Thus, we use a tensor proposed by [3, 10] to model co-occurrence of the article words. In this model, we find the co-occurred words by sliding a window across the article text. This yields to a word by word matrix for each article. By stacking all these matrices, we create a three mode tensor where the first and the second modes correspond to the words and the third mode corresponds to the articles as illustrated in Figure 4. We use this model because as shown in [1] it outperforms some state-of-the-art text based modeling in terms of classification performance and could be applied to many document and text classification tasks.
- **(Article, Domain feature) Matrix**: Another existing information in this dataset is the publisher web features in form of HTML tags. The rationale behind using these features is that different domains have different web styles. For instance trustworthy publishers like BBC and CNN tend to have standard webpages while unreliable resources often have messy webpages full of Ads, pop-ups etc. In [1], it has been shown that taking into account this information leads to a very promising classification performance. Therefore, We created a matrix out of the HTML features of the domains as demonstrated in Figure5.

Henceforth, we refer to the word tensor and publisher matrix as $\mathcal{X}_{TTA}$ and $\mathbf{X}_{TAGS}$ respectively.

6.2.2 Implementation. To capture the article representation with respect to the introduced aspects above, we use the CP/PARAFAC and SVD to decompose the $\mathcal{X}_{w\times w\times n}$ and $\mathbf{X}_{N\times P}$ into view matrices. In fact, in this case, the views are the factor matrices corresponding to the article mode and are of size $N \times r_m$ where N and r_m are the number of articles and the rank of decomposition respectively:

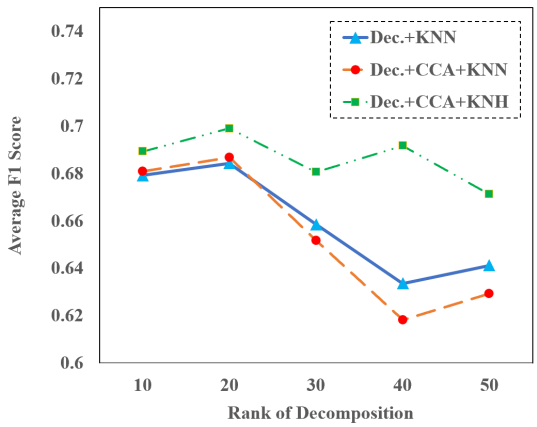
$$\mathcal{X}_{TTA} \approx \Sigma_{r=1}^{R} \mathbf{a}_r \circ \mathbf{b}_r \circ \mathbf{c}_r \tag{20}$$

$$X_{TAGS} \approx \mathbf{U}\Sigma\mathbf{V}^T \tag{21}$$

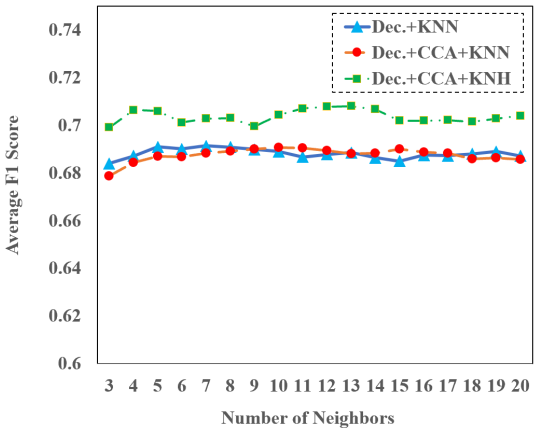
After decomposing $\mathcal{X}_{TTA}$ and $\mathbf{X}_{TAGS}$ into view matrices using CP/PARAFAC and SVD respectively, we apply the CCA on factor matrices C and U that represents the articles patterns. The result of CCA provides us with the canonical matrices where the row i of these matrices correspond to datapoints P_{1i} and P_{2i} which could be leveraged to define a line or a 2-views representation of news article i. We construct a graph such that the lines are the nodes and the edges are defined as mean distances between the lines and the points on the other lines using the equation 16. Finally, to classify articles, we leveraged the belief propagation algorithm implemented in [15] to propagate 40% of the labels throughout the KNH graph in a *semi-supervised* manner.

6.2.3 Experimental Result. To evaluate the performance of proposed KNH in comparison to classic KNN, we create a KNN graph by calculating the Euclidean distance between the rows of each view matrix separately and model the similarity of articles by taking the average pairwise distances of points. In other words, to calculate the distance between articles i and j, we calculate the pairwise Euclidean distance of rows i and j for both matrices C and U individually and then take the average of the resulted distances and consider it as the edge between node i and j in KNN graph. Moreover, to make the comparison between the KNH and KNN graphs fair enough, in another KNN model, we also project C and U using CCA into a maximally correlated space. Although this step is not required for KNN graph due to independency of views

[2] https://github.com/KaiDMML/FakeNewsNet
[3] http://bsdetector.tech/



(a) Average F1-score for 10 runs of decomposition using k=15 when modeling the articles by KNN and KNH graphs. The results suggest that KNH leads to higher performance especially when we increase the rank.



(b) Average F1-score for 10 runs of decomposition using R=20 when modeling the articles by KNN and KNH graphs. As depicted, for all number of neighbors KNH results in higher classification performance.

Figure 3: Average F1 score of KNH and KNN modeling for different ranks and number of neighbors.

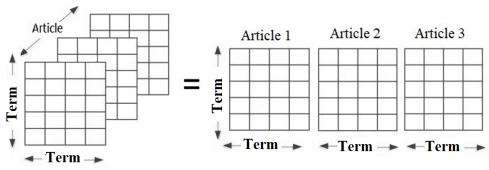


Figure 4: (Term, Term, Article) tensor to model the textual patterns form by co-occurrence of the words.

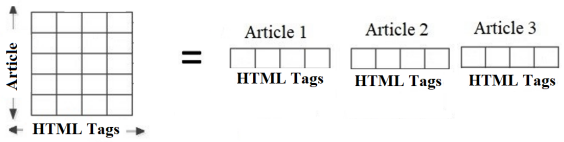


Figure 5: (Article, Domain feature) Matrix to model the publisher domain features.

in KNN modeling, we apply CCA to minimize the effect of other pre-processes in classification performance.

The average F1 score achieved by 10 runs for different ranks of decomposition and number of neighbors k are demonstrated in Figure 3. The trend of F1 score for three modelings, i.e., KNN, KNN after CCA and KNH graphs suggests that, KNH or manifold modeling of the articles using 2-flats (lines) in this case, leads to higher classification performance. As Shown, the highest performance achieved by rank 20 for all three models. Thus, we report the precision, recall, F1-score and accuracy for this R=20 and k=15 in Table. 3. Henceforth, we refer to SVD and CP as decomposition or Dec..

As reported in Table. 3, applying CCA before KNN modeling does not affect the results significantly due to independency of views in this approach. Moreover, the reported results of this table achieved by rank 20 where the KNN has the highest performance and the difference between the two models is minimum. However, this difference increases significantly when we increase the rank of decomposition which means when we capture more details of each view the manifold representation of KNH is more capable to take advantage of it.

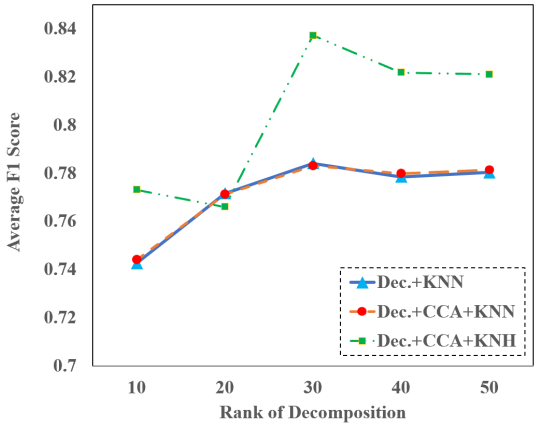
Method	Precision	Recall	F1 Score	Accuracy
Dec.+KNN	0.687±0.008	0.683±0.011	0.684±0.008	0.694±0.007
Dec.+CCA+KNN	0.691±0.007	0.682±0.018	0.686±0.011	0.697±0.011
Dec.+CCA+KNH	**0.709±0.011**	**0.720±0.016**	**0.713±0.012**	**0.719±0.011**

Table 3: Classification performance of KNH modeling against KNN modeling for R=20 and K=15 on Twitter dataset. The results suggest that regardless of whether we apply the correlation maximization on views or not the KNH outperforms the classic KNN.

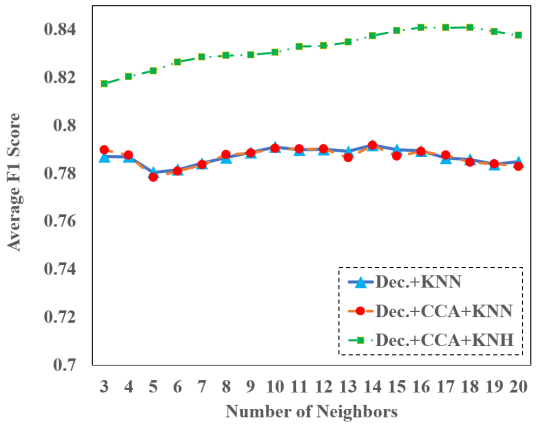
6.3 Experiment 2: Article Classification using 2-Flats (Lines) Created by User-News and Publisher-News Interactions Aspects

6.3.1 Description of Dataset and Aspects. For the second experiment, we again aim at modeling the news articles but this time using aspects other than those of previous experiment and from a different dataset to examine the efficacy of KNH on aspects of different nature. To this end, we use the FakeNewsNet dataset[16] [4] which consists of users and publisher information for news articles crawled from PolitiFact web site. The content of this website is typically shared on social media such as Twitter. The reason for using these two aspects is that in [12] the author have shown that these two aspects lead to promising result in terms of classification of articles. The details of the FakeNewsNet dataset is reported in table 5. For this experiment we use the following models as suggested in [12] to examine the efficacy of KNH modeling in comparison to classic KNN:

[4] https://github.com/KaiDMML/FakeNewsNet



(a) Average F1-score for 10 runs of decomposition and k=20 when modeling the articles by KNN and KNH graphs. As shown, KNH modeling achieves higher F1 scores in comparison to KNN modeling.



(b) Average F1-score for 10 runs of decomposition and R=30 when modeling the articles by KNN and KNH graphs. As shown, KNH modeling achieves higher F1 scores in comparison to KNN modeling.

Figure 6: Average F1 score for different rank of decomposition R and different number of neighbors K.

FakeNewsNet dataset		
Features	**Real**	**Fake**
Total news articles	432	624
Total number of tweets	116005	261262
Total news with social engagement	342	314
Total number of Users	214049	700120

Table 4: FakeNewsNet dataset description

Method	**Precision**	**Recall**	**F1 Score**	**Accuracy**
Dec.+KNN	0.836±0.004	0.742± 0.004	0.789±0.001	0.780±0.002
Dec.+CCA+KNN	0.833±0.003	0.747±0.002	0.787±0.001	0.777±0.001
Dec.+CCA+KNH	**0.875±0.002**	**0.808±0.002**	**0.839±0.001**	**0.830±0.001**

Table 5: Classification performance of KNH modeling against KNN modeling for R=30 and K=20 on FakeNewsNet dataset. The results suggest that regardless of whether we apply the correlation maximization on views or not the KNH outperforms the classic KNN.

- **User-News Interaction**: As suggested in [12] We create a matrix to model the users who tweets a specific news article. The rows of this matrix are users and the columns are the news IDs.
- **Publisher-News Interaction**: We create a matrix to model the publishers that published a specific news article. The rows of this model are the publishers and the columns are the news IDs [12].

Henceforth, we refer to the User-News interaction and the publisher-news interaction matrices as X_{UN} and X_{PN} respectively.

6.3.2 Implementation. To capture the latent representation of articles in view spaces, we first decompose the X_{UN} and X_{PN} using SVD rank r_m individually as follows:

$$X_{PN} \approx \mathbf{U_1 \Sigma_1 V_1}^T \tag{22}$$

$$X_{UN} \approx \mathbf{U_2 \Sigma_2 V_2}^T \tag{23}$$

Where X_{UN} and X_{PN} are of size $U \times N$ and $P \times N$ respectively and the V_1 and V_2 matrices are of size $N \times r_1$ and $N \times r_2$ and contain latent patterns of entities (news articles in this case). Then as explained earlier, we apply the CCA to transfer view matrices into a maximally correlated common space. Then we create a KNH graph in which the nodes are the lines or 2-flats in r_m dimentional space and the edges are defined as the mean euclidean distance between the lines and the points lie on the other lines. Finally, just like the previous experiment, we leveraged the belief propagation algorithm to propagate 40% of the ground truth in a semi-supervised manner.

6.3.3 Experimental Result. Again to compare the proposed KNH and the classic KNN graph, we follow the same strategy to calculate the similarity of article i and j. In other words, we calculate the Euclidean distance of rows i and j for both matrices V_1 and V_2 and then take the average of the resulted distances. Likewise the previous experiment, to have a fair comparison between the KNH and KNN graphs we also report the results of KNN after projection using CCA. The average F1 score achieved by 10 runs of these experiments for different ranks of decomposition are demonstrated in Figure 6.

This experiment also yields to similar results i.e., The trend of F1 score for the three different modelings, suggests that, manifold modeling of the articles using 2-flats or lines, leads to higher classification performance. As illustrated, best results achieved by rank 30 for all models. Classification metrics for R=30 and K=20 are reported in Table. 5. Like previous experiment, by increasing the rank, KNH modeling achieves higher performance than KNN graphs which again suggest that KNH is more capable of consolidating details of views.

7 CONCLUSION AND FUTURE WORK

In this work, we introduce a novel multi-view modeling of the entities (articles) by generalizing the classic KNN graph. We propose to model nodes of the graph as hyperplanes (m-flats) using datapoints derived from different views of the articles and then suggest a way to define the edges between hyperplanes. We experiment the proposed K-Nearest Hyperplane graph (KNH) on two different 2-aspect

datasets. The experimental results suggest that for different ranks and number of neighbors KNH graph outperforms the classic KNN graph. However, there are many possible directions for improving the idea of this work. Some of them are as follows:

- As discussed in background section, we can leverage parametric equations of hyperplanes for formulating and representing the m-view entities by m-flats. We experimented on 2 different 2-aspect datasets. However, by increasing the number of views we require more mathematical tools to calculate requirements of the Euclidean subspaces e.g. cross product in higher dimensional space. Unfortunately, due to the space limitation we are not able to discuss it in details. We reserve the higher view formulation of this work for future work. Moreover, as mentioned earlier, a rationale behind defining a common space for multi-view nodes in addition to a consolidate representation of the enities is to take advantage of this common space for estimating missing or unknown features that may fall into this common space. In future work, we will also explore the capability of KNH graph for prediction of missing features.
- Even though we defined the simplest way for defining the edges or multi-view similarity of nodes in KNH graphs in this work, we are interested in defining more insightful edges between the nodes by probably merging the capabilities of hypergraphs that take into account higher order relations between the nodes and the advantages of multi-aspect nodes of this work. We reserve the study and formulation of more meaningful edges for future work.

8 ACKNOWLEDGEMENTS

Research was supported by a UCR Regents Faculty Fellowship, a gift from Snap Inc., the Department of the Navy, Naval Engineering Education Consortium under award no. N00174-17-1-0005, and the National Science Foundation Grant no. 1901379. The authors would like to thank Rutuja Gurav for her invaluable help with the proofreading of the paper. Any opinions, findings, and conclusions or recommendations expressed in this material are those of the author(s) and do notnecessarily reflect the views of the funding parties.

REFERENCES

[1] Sara Abdali, Neil Shah, and Evangelos E. Papalexakis. 2020. FHiJoD: Semi-Supervised Multi-aspect Detection of Misinformation using Hierarchical Joint Decomposition. *arXiv preprint arXiv: arXiv:2005.04310v1* (2020).

[2] T. G. Kolda B. W. Bader et al. 2015. Matlab tensor toolbox version 2.6. Available online.

[3] Gisel G. Bastidas, Sara Abdali, Neil Shah, and Evangelos E. Papalexakis. 2018. Semi-supervised Content-based Detection of Misinformation via Tensor Embeddings. In *Advances in Social Networks Analysis and Mining (ASONAM), 2018 IEEE/ACM International Conference on*. IEEE, 322 – 325.

[4] K. G. Binmore. 1981. In *The Foundations of Topological Analysis: A Straightforward Introduction*. Cambridge University Press.

[5] Ian Cook. 2002. Oxford Dictionary of Statistics. In *Oxford University Press*. 104.

[6] Thomas G. Dietterich. 2000. Ensemble Methods in Machine Learning. In *Multiple Classifier Systems*. Springer Berlin Heidelberg, Berlin, Heidelberg, 1–15.

[7] Giorgio Gallo, Giustino Longo, Stefano Pallottino, and Sang Nguyen. 1993. Directed Hypergraphs And Applications. *Discrete Applied Mathematics* 42 (04 1993), 177–201. https://doi.org/10.1016/0166-218X(93)90045-P

[8] Jiawei Han, Micheline Kamber, and Jian Pei. 2011. *Data Mining: Concepts and Techniques* (3rd ed.). Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[9] R. A. Harshman. 1970. Foundations of the PARAFAC procedure: Models and conditions for an" explanatory" multi-modal factor analysis. *UCLA Working Papers in Phonetics* 16, 1 (1970), 84.

[10] Seyedmehdi Hosseinimotlagh and Evangelos E. Papalexakis. 2017. Unsupervised Content-Based Identification of Fake News Articles with Tensor Decomposition Ensembles.

[11] Y. Huang, Q. Liu, F. Lv, Y. Gong, and D. N. Metaxas. 2011. Unsupervised Image Categorization by Hypergraph Partition. *IEEE Transactions on Pattern Analysis and Machine Intelligence* 33, 6 (June 2011), 1266–1273. https://doi.org/10.1109/TPAMI.2011.25

[12] S. Wang K. Shu, A. Sliva and H. Liu. 2019. Beyond news contents: the role of social context for fake news detection. In *WSDM '19 Proceedings of the Twelfth ACM International Conference on Web Search and Data Mining*. 312–320.

[13] S. Wang D. Lee K. Shu, L. Cui and H. Liu. 2019. dEFEND: Explainable Fake News Detection. In *Proceedings of 25th ACM SIGKDD international conference on Knowledge discovery and data mining*.

[14] S. Wang J. Tang K. Shu, A. Sliva and H. Liu. 2017. Fake News Detection on Social Media: A Data Mining Perspective. *ACM SIGKDD Explorations Newsletter* 19, 1 (2017), 22–36.

[15] Danai Koutra, Tai-You Ke, U. Kang, Duen Chau, Hsing-Kuo Pao, and Christos Faloutsos. 2011. Unifying Guilt-by-Association Approaches: Theorems and Fast Algorithms. In *Machine Learning and Knowledge Discovery in Databases (ECML/PKDD)*. Lecture Notes in Computer Science, Vol. 6912. 245–260.

[16] S. Wang D. Lee K.Shu, D. Mahudeswaran and H. Liu. 2018. FakeNewsNet: A Data Repository with News Content, Social Context and Dynamic Information for Studying Fake News on Social Media. *arXiv preprint arXiv:1809.01286* (2018).

[17] Srijan Kumar and Neil Shah. 2018. False information on web and social media: A survey. *arXiv preprint arXiv:1804.08559* (2018).

[18] Xi Li, Yao Li, Chunhua Shen, Anthony Dick, and Anton Hengel. 2013. Contextual Hypergraph Modeling for Salient Object Detection. *Proceedings of the IEEE International Conference on Computer Vision* (10 2013). https://doi.org/10.1109/ICCV.2013.413

[19] Xi Li, Yao Li, Chunhua Shen, Anthony Dick, and Anton Hengel. 2013. Contextual Hypergraph Modeling for Salient Object Detection. *Proceedings of the IEEE International Conference on Computer Vision* (10 2013). https://doi.org/10.1109/ICCV.2013.413

[20] Xi-Lin Li, Matthew Anderson, and Tülay Adalı. 2010. Second and Higher-Order Correlation Analysis of Multiple Multidimensional Variables by Joint Diagonalization. In *Latent Variable Analysis and Signal Separation*, Vincent Vigneron, Vicente Zarzoso, Eric Moreau, Rémi Gribonval, and Emmanuel Vincent (Eds.). Springer Berlin Heidelberg, Berlin, Heidelberg, 197–204.

[21] Yong Luo, Dacheng Tao, Yonggang Wen, Kotagiri Ramamohanarao, and Chao Xu. 2015. Tensor Canonical Correlation Analysis for Multi-View Dimension Reduction. *IEEE Transactions on Knowledge and Data Engineering* 27 (02 2015). https://doi.org/10.1109/TKDE.2015.2445757

[22] Evangelos E. Papalexakis, Christos Faloutsos, and Nicholas D. Sidiropoulos. 2016. Tensors for Data Mining and Data Fusion: Models, Applications, and Scalable Algorithms. *ACM Trans. Intell. Syst. Technol.* 8, 2, Article 16 (Oct. 2016), 44 pages. https://doi.org/10.1145/2915921

[23] N.D. Sidiropoulos, Lieven De Lathauwer, Xiao Fu, Kejun Huang, Evangelos Papalexakis, and Christos Faloutsos. 2016. Tensor Decomposition for Signal Processing and Machine Learning. *IEEE Transactions on Signal Processing* PP (07 2016). https://doi.org/10.1109/TSP.2017.2690524

[24] Lifan Su, Yue Gao, Xibin Zhao, Hai Wan, Ming Gu, and Jiaguang Sun. 2017. Vertex-Weighted Hypergraph Learning for Multi-View Object Classification. 2779–2785. https://doi.org/10.24963/ijcai.2017/387

[25] Liang Sun, Shuiwang Ji, and Jieping Ye. 2008. Hypergraph Spectral Learning for Multi-label Classification. *Proceedings of the Fourteenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 668–676. https://doi.org/10.1145/1401890.1401971

[26] M. Wang, X. Liu, and X. Wu. 2015. Visual Classification by ℓ_1 -Hypergraph Modeling. *IEEE Transactions on Knowledge and Data Engineering* 27, 9 (Sep. 2015), 2564–2574. https://doi.org/10.1109/TKDE.2015.2415497

[27] Liang Wu, Jundong Li, Xia Hu, and Huan Liu. 2017. Gleaning wisdom from the past: Early detection of emerging rumors in social media. In *Proceedings of the 2017 SIAM International Conference on Data Mining*. SIAM, 99–107.

[28] Liang Wu and Huan Liu. 2018. Tracing fake-news footprints: Characterizing social media messages by how they propagate. In *Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*. ACM, 637–645.

[29] J. Yu, D. Tao, and M. Wang. 2012. Adaptive Hypergraph Learning and its Application in Image Classification. *IEEE Transactions on Image Processing* 21, 7 (July 2012), 3262–3272. https://doi.org/10.1109/TIP.2012.2190083

[30] Yunqian M. Zhang C. 2000. Ensemble Machine Learning. In *Ensemble Machine Learning*. Springer, Boston, MA, Boston, MA.

[31] Dengyong Zhou, Jiayuan Huang, and Bernhard Schölkopf. 2006. Learning with Hypergraphs: Clustering, Classification, and Embedding. *Advances in Neural Information Processing Systems 19: Proceedings of the 2006 Conference, 1601-1608 (2007)* 19, 1601–1608.