Hypernode Graphs for Learning from Binary Relations between Groups in Networks

Thomas Ricatte
Inria
Lille, France
thomas@ricatte.fr

Rémi Gilleron
Université de Lille
Villeneuve d'Ascq, France
remi.gilleron@univ-lille3.fr

Marc Tommasi
Université de Lille
Villeneuve d'Ascq, France
marc.tommasi@univ-lille3.fr

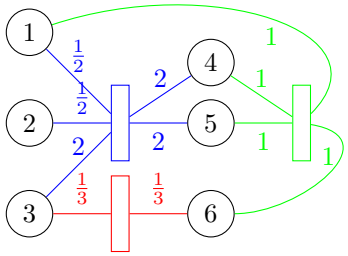
Abstract

The aim of this paper is to propose methods for learning from interactions between groups in networks. Hypernode graphs were introduced in [15] as a formal tool able to model group interactions. Spectral learning algorithms were defined allowing to infer node ratings or node labelings. As a proof of concept, we have modeled multiple players games with hypernode graphs and have defined skill rating algorithms competitive with specialized algorithms. In this paper, we explore theoretical issues related to hypernode graphs. We first demonstrate that hypernode graph kernels strictly generalize over graph kernels and hypergraph kernels. We then proceed to prove that hypernode graphs correspond to signed graphs such that the matrix $D - W$ is positive semidefinite. It should be noted that homophilic relations between groups may lead to non homophilic relations between individuals. We finally define the notion of connected hypernode graphs and a resistance distance for connected hypernode graphs.

1 Introduction

Networks are commonly modeled by graphs where nodes correspond to individual objects and edges correspond to binary relationships between individuals. This is for instance the case for social networks with friendships between users, or computer networks with connections between machines. In many cases, only interactions between groups are observed without any information on individuals besides group membership. This is for instance the case for online games where only game outcomes for games between teams of multiple players are given. Also, group interactions are observed when dealing with clusters in computer networks or communities in social networks. However in such contexts it might be still interesting to evaluate individual objects, like for instance players skills, CPU loads or users profiles and properties.

In [15], we introduced hypernode graphs as a formal tool able to model group interactions. We defined a *hypernode graph* to be a set of hyperedges. A hyperedge is a pair of disjoint hypernodes, where a hypernode is a set of nodes. Every node of a hypernode is given a non negative real-valued weight and weights in a hyperedge satisfy an equilibrium condition. Roughly speaking, a hypernode models a group, a hyperedge models a relation between two groups, and individual weights correspond to the contribution of each individual to the relation between the two groups. An example of hypernode graph is shown in Figure 1. It is easy to show that there is a one to



$h_{blue} = \{\{1,2,3\},\{4,5\}\}$

$h_{red} = \{\{3\},\{6\}\}$

$h_{green} = \{\{4,5\},\{1,6\}\}$

Figure 1: A hypernode graph with three hyperedges. A hyperedge is represented by a rectangle. Nodes in one of the two hypernodes are connected to the same side of the rectangle.

one correspondence between undirected graphs and hypernode graphs where all hypernodes are singleton sets.

In many applications, real-valued functions that are defined on nodes and on sets of nodes coincidentally, play an important role. In the examples given above, such functions measure CPU or cluster loads, individual players or team skills, users or communities ratings. The hypernode graph model assumes that the function value on a set of nodes is a linear weighted sum of node function values. For instance, in multiple player games, the team performance value is assumed to be the weighted sum of player skills [6, 8]. Then, a learning task is to infer node function values over a hypernode graph. We consider a semi-supervised setting where function values are given for only a few nodes or a few hypernodes.

For semi-supervised learning on hypernode graphs, we assume that connected hypernodes tend to have similar values. For real-valued node functions, the notion of gradient is a mathematical tool that allows to measure this similarity. Therefore, an appropriate notion of gradient for hypernode graphs was defined in [15]. And if G is the gradient of a hypernode graph $\mathbf{h}$, $f^T G^T G f$ measures the smoothness of a real-valued node function f. The matrix $G^T G$ is defined to be the Laplacian of the hypernode graph h. And, because, $G^T G$ is positive semidefinite, spectral learning algorithms (like [17, 19, 21]) can be applied for learning in hypernode graphs. As a proof-of-concept, we have shown how to use hypernode graphs and the associated spectral theory to infer skill ratings in the case of multiple players games.

One question raised by the introduction of hypernode graphs is whether problems on hypernode graphs can be solved using graphs, i.e. whether hypernode graphs strictly generalize over graphs. Before, let us recall that a previously defined generalization of graphs is hypergraphs where a hyperedge is a set of nodes. Hypergraphs have been studied from a machine learning perspective and have been used in many applications [11, 18, 4]. Hypergraphs do not really generalize over graphs from a spectral point of view. Indeed, it has been proved in [1] that every hypergraph Laplacian introduced so far corresponds to a graph Laplacian using an adequate graph construction. Also, it should be noted that hypergraphs model n-ary relations between nodes but fail to model relations between sets of nodes. Here, we show that hypernode graphs strictly extend the expressive power of graphs with respect to the class of Laplacians they define. Indeed, we show that that the class of hypernode graph Laplacians strictly contains the class of graph Laplacians and the class of graph kernels, where a graph kernel is defined to be the Moore-Penrose pseudoinverse of a graph Laplacian. As an intermediate result, we observe that a convex linear combination of graph kernels (used in multiple graph kernel learning as in [2]) is a hypernode graph kernel. Moreover, we show that no graph construction allows to define a set of smooth functions over the graph which coincides with the set of smooth functions over a given hypernode graph.

Another argument showing that hypernode graphs strictly generalize over graphs is that, for every hypernode graph, we can construct a signed graph with adjacency matrix W such that $D - W$ is the Laplacian of the hypernode graph. As a consequence of this result, we get that homophilic relations between hypernodes may lead to non homophilic relations between nodes because of the possible negative weights. Let us recall that, for arbitrary signed graphs, the matrix $D - W$ can be indefinite. Thus, many works have studied the notion of Laplacian for signed graphs. Herbster in [9] propose to consider the class of symmetric diagonally dominant matrices with non negative

diagonal entries. While others ([10, 7, 14]) have used a modified definition of the Laplacian to obtain a positive semidefinite Laplacian. But the graph interpretation of the modified Laplacian is not preserved. Koren et al in [13] have considered the class of signed graphs such that $D - W$ is positive semidefinite, but this class was not studied from machine learning perspective. Thus, here we consider the same class of signed graphs and obtain an alternative representation and an interpretation using hypernode graphs.

Last, we study the definition of a distance between nodes in hypernode graphs. Indeed, it has been shown that, for connected graphs, a resistance distance can be defined from the graph kernel and that it can be expressed in term of the commute time distance in the graph [12, 5]. We define a pseudo-distance between nodes of a hypernode graph using the hypernode graph kernel. We show that it is a distance for a well chosen notion of connected hypernode graphs. But, it should be noted that we leave open the question of finding an algorithmic definition of connected components in hypernode graphs. Also, we introduce a diffusion operator for hypernode graphs and we show that the distance can be expressed from the differences of potentials. As a special case, we get the result on the commute time distance for graphs. But, when the hypernode graph is not a graph, the question of finding an interpretation of the distance between two nodes with random walks is left open because negative terms are involved.

2 Hypernode Graphs and Application to Skill Rating

A *hypernode graph* $\mathbf{h} = (V, H)$ is a set of nodes V with $|V| = n$ and a set of hyperedges H with $|H| = p$. A hypernode is a set of nodes and a hyperedge $h \in H$ is an unordered pair $\{s_h, t_h\}$ of two non empty and disjoint hypernodes. Each hyperedge $h \in H$ has a weight function w_h mapping every node i in $s_h \cup t_h$ to a positive real number $w_h(i)$. Each weight function w_h of $h = \{s_h, t_h\}$ must satisfy the *Equilibrium Condition* defined by $\sum_{i \in t_h} \sqrt{w_h(i)} = \sum_{i \in s_h} \sqrt{w_h(i)}$. We can extend the weight function w_h over V by defining $w_h(i) = 0$ for every node i not in $s_h \cup t_h$. An example of hypernode graph is shown in Figure 1. The blue hyperedge h_{blue} links the sets $\{1, 2, 3\}$ and $\{4, 5\}$. The weights of h_{blue} satisfy the Equilibrium condition: $\sqrt{1/2} + \sqrt{1/2} + \sqrt{2} = \sqrt{2} + \sqrt{2}$. The red hyperedge h_{red} links the two singleton sets $\{3\}$ and $\{6\}$. The weights of h_{red} are equal because of the Equilibrium condition. The hyperedge h_{red} can be viewed as an edge with edge weight $1/3$.

We are mainly interested in evaluating real-valued functions on nodes and on hypernodes. We assume a weighted linear model such that any real-valued node function f can be extended to hypernodes in $u \in h = \{s_h, t_h\}$ by $f(u) = \sum_{i \in u} f(i)\sqrt{w_h(i)}$. The homophilic assumption for hypernode graphs states that connected hypernodes tend to have similar values. The assumption can be measured using an appropriate gradient definition. The *unnormalized gradient* of a hypernode graph $\mathbf{h} = (V, H)$ is defined to be the linear application, denoted by grad, that maps every real-valued node function f into a real-valued hyperedge function $\mathrm{grad}(f)$ defined, for every hyperedge $h = \{s_h, t_h\}$ in H, by

$$\mathrm{grad}(f)(h) = \sum_{i \in t_h} f(i)\sqrt{w_h(i)} - \sum_{i \in s_h} f(i)\sqrt{w_h(i)} \ ,$$

where an arbitrary orientation of the hyperedges has been chosen. Because of the Equilibrium Condition, the gradient of every constant node function is the zero-valued hyperedge function. Let us denote by $G \in \mathbb{R}^{p \times n}$ the matrix of grad, the square $n \times n$ real valued matrix $\Delta = G^T G$ is defined to be the *unnormalized Laplacian* of the hypernode graph $\mathbf{h}$. When the hypernode graph is a graph (all hypernodes are singleton sets), then the hypernode graph Laplacian is equal to the graph Laplacian. The Laplacian Δ does not depend on the arbitrary orientation of the hyperedges used for defining the gradient. We define the *smoothness* of a real-valued node function f over a hypernode graph $\mathbf{h}$ to be $\Omega(f) = f^T \Delta f$. Last, we define the *hypernode graph kernel* of a hypernode graph $\mathbf{h}$ to be the Moore-Penrose pseudoinverse $\Delta^\dagger$ of the hypernode graph Laplacian Δ. We have proved in [15] that

Proposition 1. *The class of hypernode graph Laplacians is the class of symmetric positive semidefinite real-valued matrices M such that $\mathbf{1} \in \mathrm{Null}(M)$, where $\mathrm{Null}(M)$ is the null space of M.*

Because a hypernode graph Laplacian is positive semidefinite, we can leverage the spectral learning algorithms defined in [17], [19], [21] from graphs to hypernode graphs. In the following we will

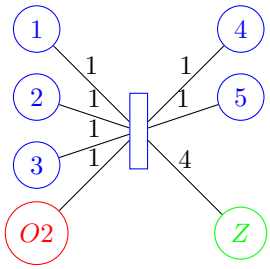


Figure 2: Hyperedge h for a game $A = \{1, 2, 3\}$ against $B = \{4, 5\}$ where B wins with score 2. The outcome node $O2$ corresponds to the score 2. All weights are set to 1 except the weight for Z chosen in order to ensure the Equilibrium condition.

sketch how to use learning algorithms for hypernode graphs for the skill rating problem in multiple players games. A more detailed presentation can be found in [15].

Let us consider a set of players $P = \{1, \ldots, n\}$ and a set of games Γ. Each game is between two teams of multiple players and we assume that each player contributes to its team with a positive real value. We also suppose given the game outcome of every game in Γ. Then, the construction of the hypernode graph $\mathbf{h} = (V, H)$ is as follows. The set of nodes is defined to be the set of node players $\{1, \ldots, n\}$ plus a finite set of outcome nodes (one per possible score) plus one lazy node. For every game, we define a hyperedge as follows :

- one hypernode contains the node players for the losing team and one outcome node corresponding to the game outcome;
- one hypernode contains the node players for the winning team and the lazy node;
- the weights for player nodes are chosen to be the player contributions, the weight of the outcome node is set to 1, the weight of the lazy node is chosen in order to satisfy the Equilibrium condition.

Note that if the game outcome is a draw, any of the two teams can be chosen as the losing team and the game outcome node corresponds to a null score. We show a hyperedge in Figure 2 and use it to illustrate how it allows to model player skills. Let us consider a real-valued node function f over the hyperedge h, and let us fix $f(O2) = 2$ and $f(Z) = 0$. Then the homophilic property states that $f(1) + f(2) + f(3) + 2$ is close from $f(4) + f(5)$ which expresses the expected relations between player skills given the game outcome. And, the homophilic assumption can be measured using the smoothness measure Ω. Indeed, given the hypernode graph $\mathbf{h}$ constructed as above with Laplacian Δ, the *skill rating problem* is equivalent to find the real-valued node function s solving

$$\begin{aligned} \text{minimize } s^T \Delta s \quad \text{subject to } & s(Z) = 0 \text{ (for the lazy node)} \\ & s(O_j) = o_j \text{ (for outcome nodes)} \end{aligned} \tag{1}$$

Moreover, in [15], we have introduced a regularization term to limit the spread of the function s to ensure that, when the number of games is small, player skills remain comparable. We have also shown that the regularization term can be modeled in the hypernode graph with an ad hoc construction. Thus, the skill rating problem can be modeled as in Equation 1 with the modified hypernode graph. We have considered two methods: the first one is to use the semi-supervised learning algorithm presented in [21] because the Laplacian Δ of a hypernode graph is positive semidefinite; the second one is to infer player nodes scores from lazy nodes scores and outcome nodes scores using a regression support vector machine with the hypernode graph kernel $\Delta_\mu^\dagger$.

We have applied these two algorithms in a batch setting for the skill rating problem over simple tennis games, double tennis games and online games (Halo2 dataset). Experimental results are given in [15]. They show that we obtain very competitive results compared with specialized algorithms such as Elo duelling [6] and TrueSkill [8].

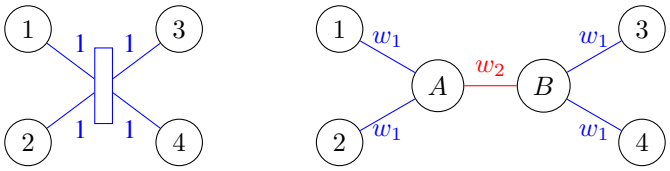


Figure 3: A hypernode graph and a candidate star expansion.

3 Hypernode Graphs, Graphs and Signed Graphs

3.1 The Class of Hypernode Graphs Laplacians

As a consequence of Proposition 1, it is easy to show that the class of hypernode graph Laplacians

- is closed by convex linear combination,
- is closed by pseudoinverse,
- and strictly contains the class of graph Laplacians and the class of graph kernels, where a graph kernel is the Moore-Penrose pseudoinverse of a graph Laplacian.

As a consequence, we get that a convex linear combination of graph kernels is a hypernode graph kernel (also a hypernode graph Laplacian). Linear combinations of graph kernels have been used for semi-supervised learning [2]. The fact the class of hypernode graph Laplacians contains strictly the class of graph Laplacians does not allow us to claim that hypernode graphs provide a gain of expressiveness. Indeed, it could be the case that through a well chosen graph construction, the set of smooth functions defined on hypernodes could be made to coincide exactly with the set of smooth functions defined on graphs. This is, for instance, the case for hypergraphs. Indeed, it has been shown in [1] that all hypergraph Laplacians defined so far – among them the Laplacians Δ_{B} from [3], Δ_{R} from [16] and Δ_{ZHS} from [20] – can be defined as (restrictions of) graph Laplacians using a graph construction such as the clique expansion (where each hyperedge is replaced by a clique graph with uniform weights) or the star expansion (where, for every hyperedge, a new node is added and is linked with all the nodes in the hyperedge).

While one can think of similar constructions for the case of hypernode graphs, we prove that there does not exist a graph expansion of a hypernode graph which defines the same set of smooth functions over the original set of nodes. The proof is based on the very simple hypernode graph shown in Figure 3. We show by contradiction that there does not exists a finite graph whose node set contains $\{1, 2, 3, 4\}$ which can express that $(f(1) + f(2) - f(3) - f(4))^2 = 0$.

To conclude this section, we propose a conjecture

Conjecture 1. *The class of hypernode graph Laplacians is the smallest class closed by convex linear combinations which contains all graph kernels.*

The difficult part is to prove that every hypernode graph Laplacian is a convex linear combination of graph kernels. This is because a graph kernel, which is defined to be the pseudoinverse of a graph Laplacian, has no simple analytic form.

3.2 L-equivalent Hypernode Graphs and Signed Graphs

Let us consider a symmetric positive semidefinite real-valued matrix M such that $\mathbf{1} \in \mathrm{Null}(M)$, M is the Laplacian of a hypernode graph by Proposition 1. Indeed, let us consider a square root decomposition $G^T G$ of M. Then, for each line of G, we can define a hyperedge $h = \{s_h, t_h\}$ where nodes in s_h (respectively in t_h) have positive (respectively negative) values in the line. The weights are chosen to be square roots of absolute values in the line. The equilibrium condition is satisfied because $\mathbf{1} \in \mathrm{Null}(M)$. This leads to a hypernode graph $\mathbf{h}$ with Laplacian M. But, as the square root decomposition is not unique, there are several hypernode graphs with the same Laplacian that we called *L-equivalent* . Examples of L-equivalent hypernode graphs are given in Figure 4. In order to study the L-equivalent relation and to obtain results showing whether a hypernode graph is L-equivalent to a graph, we first show that hypernode graphs are related to signed graphs.

$$G = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix} \qquad G = \begin{pmatrix} -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & \sqrt{2} & 0 \\ -\frac{\sqrt{2}}{2} & -\frac{\sqrt{2}}{2} & 0 & \sqrt{2} \end{pmatrix}$$

Figure 4: two L-equivalent hypernode graphs with their respective gradient which are square root of the Laplacian Δ shown in Figure 5.

$$\Delta = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 1 & 1 & -1 & -1 \\ -1 & -1 & 2 & 0 \\ -1 & -1 & 0 & 2 \end{pmatrix}; W = \begin{pmatrix} 0 & -1 & 1 & 1 \\ -1 & 0 & 1 & 1 \\ 1 & 1 & 0 & 0 \\ 1 & 1 & 0 & 0 \end{pmatrix}; D = \begin{pmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 2 & 0 \\ 0 & 0 & 0 & 2 \end{pmatrix}.$$

Figure 5: Laplacian matrix, pairwise weight matrix, and degree matrix for the two L-equivalent hypernode graphs shown in Figure 4.

It is known that the Laplacian matrix Δ of a graph can be written $D - W$ where W is the adjacency matrix of the graph and D is the corresponding degree matrix. Let us consider a hypernode graph $\mathbf{h}$ with Laplacian Δ, we define the *pairwise weight matrix* W of $\mathbf{h}$ by $W_{i,j} = -\Delta_{i,j}$ if $i \neq j$, and 0 otherwise. Note that the pairwise weight matrix of a hypernode graph which is a graph coincides with the adjacency matrix. Let us define the degree of a node i to be $\deg(i) = \sum_{j \in V} W_{i,j}$ and let us denote by D the diagonal matrix of all $\deg(i)$. Then, because of the property $\mathbf{1} \in \mathrm{Null}(M)$ in Proposition 1, it is immediate that, for every i, $\Delta_{i,i} = \sum_{j \in V} W_{i,j}$. As a consequence, we have

Proposition 2. *Let $\mathbf{h} = (N, H)$ be a hypernode graph, let W be the pairwise weight matrix of $\mathbf{h}$, and let D be the diagonal degree matrix of $\mathbf{h}$. Then, the Laplacian of $\mathbf{h}$ is $\Delta = D - W$.*

An example is shown in Figure 5. As a consequence of the above proposition, we can leverage the pairwise weight matrix to characterize L-equivalent hypernode graphs by

Proposition 3. *Two hypernode graphs are L-equivalent if and only if they have the same pairwise weight matrix. Two L-equivalent hypernode graphs have the same degree matrix.*

The pairwise weight matrix W contains in general negative weights and can be thus interpreted as the adjacency matrix of a signed graph. Therefore, we define the *reduced signed graph* of a hypernode graph $\mathbf{h}$ to be the signed graph $\tilde{\mathbf{g}}$ with adjacency matrix W. An example is shown in Figure 6. Then, we can show that

- the reduced signed graph of a graph g (viewed as a hypernode graph) is the graph g,
- a hypernode graph $\mathbf{h}$ is L-equivalent to a graph if and only if its reduced signed graph is a graph,
- a signed graph with adjacency matrix W is the reduced signed graph of a hypernode graph if and only if the matrix $D - W$ is positive semidefinite.

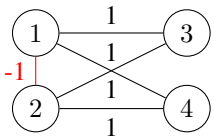


Figure 6: The reduced signed graph with adjacency matrix W of Figure 5 of the two L-equivalent hypernode graphs shown in Figure 4.

The definition of the pairwise weight matrix may seem ad hoc to mimic the definition of the Laplacian in the graph case. But, we can show that the pairwise weight matrix can be defined directly from the hypernode graph using the following formula

$$\forall i \neq j, \quad W_{i,j} = \sum_{h \in H} P(h,i,j)\sqrt{w_h(i)}\sqrt{w_h(j)} \ , \tag{2}$$

where $P(h,i,j) = 1$ if i and j belongs to two different ends of h, $P(h,i,j) = -1$ if i and j belongs to the same end of h, and 0 otherwise. And the formula can be interpreted as follows. The pairwise weight $W_{i,j}$ is computed as a sum over all the hyperedges, then it can be understood as an aggregation of all the information over hyperedges involving the nodes i and j. For every term in the sum, the quantity $\sqrt{w_h(i)}$ can be viewed as the cost of entering the hyperedge h at node i, the quantity $\sqrt{w_h(j)}$ as the cost of exiting the hyperedge h at node j, and $P(h,i,j)$ as a sign depending whether i and j are in the same end or in different ends of the hyperedge.

4 Resistance Distance and Random Walks in Hypernode Graphs

In this section, we study whether a distance can be defined between nodes of a hypernode graph and how such a distance can be interpreted in the hypernode graph. Throughout the section, we consider a hypernode graph $\mathbf{h} = (N, H)$ with Laplacian Δ.

4.1 Defining a distance in Hypernode Graphs

Let $\Delta^\dagger$ be the Moore-Penrose pseudo-inverse of Δ, let us define d by, for every i, j in N,

$$d(i,j) = \left(\sqrt{\Delta^\dagger_{i,i} + \Delta^\dagger_{j,j} - 2\Delta^\dagger_{i,j}}\right) \ . \tag{3}$$

Because $\Delta^\dagger$ is symmetric positive semidefinite, we have

Proposition 4. *d defines a pseudometric on $\mathbf{h}$, i.e., it is positive, symmetric and satisfies the triangle inequality.*

For the pseudometric d to be a distance, d should satisfy also the coincidence axiom: $d(i,j) = 0 \Rightarrow i = j$. This is not true in general as $d(1,2) = 0$ for the hypernode graph in Figure 3. The intuition is that the nodes 1 and 2 can not be distinguished because the smoothness condition is on the sum $f(1) + f(2)$. Nevertheless we can show that

Proposition 5. *When $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$, d defines a metric (or distance) on $\mathbf{h}$.*

Let us recall that, in the graph case, Proposition 4 holds and that Proposition 5 holds when the graph is connected. We will discuss connectivity properties for hypernode graphs in the next section but let us note that the property $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$ can be seen as an algebraic definition of a connected graph. Let us also note that, in the graph case, for connected graphs, both d and d^2 are metrics while, for hypernode graphs, d^2 does not satisfy the triangle inequality even if $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$.

4.2 Diffusion in Hypernode Graphs

Let us suppose that $\mathbf{h}$ satisfies $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$. Our goal is to study whether d^2 can be written in terms of a diffusion function in the hypernode graph. First, let us consider the Poisson equation $\Delta f = \mathbf{In}$ that models the diffusion of an input charge $\mathbf{In}$ through a system associated with the Laplacian operator Δ. Let us consider a node j called *sink* node, we consider the input function $\mathbf{In}_j$ defined by $\mathbf{In}_j(j) = \deg(j) - \mathrm{Vol}(\mathbf{h})$ and $\mathbf{In}_j(i) = \deg(i)$ if $i \neq j$. We can prove that

Lemma 1. *The solutions of $\Delta f = \mathbf{In}_j$ are the functions $f = \mu\mathbf{1} + \Delta^\dagger \mathbf{In}_j$ where $\mu \in \mathbb{R}$.*

The proof is omitted. It is based on properties of pseudoinverse matrices and on the hypothesis $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$. Then, for every pair of nodes (k,ℓ) in N, we define $\mathrm{V}_j(k,\ell)$ as the difference of potential between k and ℓ, i.e., we define $\mathrm{V}_j(k,\ell) = f(k) - f(\ell)$ where f is a solution of the

Poisson equation $\Delta f = \mathbf{In}_j$. Lemma 1 allows to show that the definition of $\mathrm{V}_j(k,\ell)$ does not depend on the choice of the solution of the Poisson equation and that an equivalent definition is

$$\mathrm{V}_j(k,\ell) = (\mathbf{e}_k - \mathbf{e}_\ell)^T \Delta^\dagger \mathbf{In}_j \ ,$$

where $\mathbf{e}_i$ is the unit vector with 1 in component i. We can now relate the distance d^2 and the diffusion potential V by

Proposition 6. *For every i, j in N, we have* $\mathrm{Vol}(\mathbf{h})d^2(i,j) = \mathrm{V}_j(i,j) + \mathrm{V}_i(j,i)$*, where* $\mathrm{V}_j(i,i) = 0$*, and for $i \neq j$,*

$$\mathrm{V}_j(i,j) = \sum_{h|i\in h} \frac{w_h(i)}{\deg(i)} \left[1 + \sum_{k\in h, k\neq i} P(h,i,k) \sqrt{\frac{w_h(k)}{w_h(i)}} \mathrm{V}_j(k,j) \right] \ . \tag{4}$$

We omit the proof by lack of space.

4.3 Connected Components and Random Walks in Hypernode Graphs

We have seen that in order for d to be a metric condition $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$ should be satisfied. By analogy with the case of graphs, this condition can be viewed as a definition of a connected hypergraph. But, so far, we have not found an algebraic definition of connected components in hypernode graphs.

Another interesting issue is the relation between the distance d^2 and random walks in hypernode graphs. First, let us consider a hypernode graph $\mathbf{h}$ where all hypernodes are singleton sets. The hypernode graph can be viewed as a graph and every hyperedge h that contains a node i is an edge $\{\{i\},\{k\}\}$ with $k \in N$. Thus, we have $w_h(i) = w_h(k) = W_{i,k}$ and Equation (4) can be rewritten as

$$\mathrm{V}_j(i,j) = \sum_{k\in N} \frac{W_{i,k}}{\deg(i)} (1 + \mathrm{V}_j(k,j)) \ .$$

Thus, $\mathrm{V}_j(i,j)$ can be interpreted as the hitting-time distance from i to j (average number of steps needed by a random walker to travel from i to j). Consequently, Proposition 6 states that, when $\mathbf{h}$ is a graph, the distance $d^2(i,j)$ between two nodes i and j is equal to the commute-time distance between i and j divided by the overall volume, which is already known [12, 5].

Now, let us consider the general case, and let us define $p(h|i) = \frac{w_h(i)}{\deg(i)}$ and $p(k|h,i) = P(h,i,k)\sqrt{\frac{w_h(k)}{w_h(i)}}$. Then, we can rewrite Equation (4) as

$$\mathrm{V}_j(i,j) = \sum_{h|i\in h} p(h|i) \left[1 + \sum_{k\in h, k\neq i} p(k|h,i) \mathrm{V}_j(k,j) \right] \ .$$

The term $p(h|i)$ can be interpreted as a jumping probability from node i to the hyperedge h because $p(h|i)$ is non-negative and $\sum_h p(h|i) = 1$. But, while we have $\sum_n p(k|h,i) = 1$, the term $p(k|h,i)$ is negative as soon as i and k belong to the same end of h. This prevents us from interpreting this quantity as a jumping probability from node i to node k with the hyperedge h. Therefore, there is no easy interpretation of the distance d^2 in terms of random walks in the hypernode graph because of negative values in the expression of $d^2(i,j)$.

5 Conclusion

We have introduced a model for learning from binary relations between groups in networks. We have defined a spectral theory allowing to model homophilic relations between groups assuming an additive model for individual rates. We can also model dominance relations by adding nodes as shown in the skill rating problem. We hope that the model will open the way to solving new learning problems in networks. From the theoretical point of view, many issues remain to be studied such as directed hypernode graphs and the definition of cuts among others.

References

[1] S. Agarwal, K. Branson, and S. Belongie. Higher Order Learning with Graphs. In *Proceedings of the 23rd International conference on Machine learning (ICML-06)*, pages 17–24, 2006.

[2] A. Argyriou, M. Herbster, and M. Pontil. Combining Graph Laplacians for Semi-Supervised Learning. In *Proceedings of the 19th conference Annual Conference on Neural Information Processing Systems (NIPS-05)*, pages 67–74, Cambridge, MA, 2006. MIT Press.

[3] M. Bolla. Spectra, euclidean representations and clusterings of hypergraphs. *Discrete Mathematics*, 117(1):19–39, 1993.

[4] J. Cai and M. Strube. End-to-end coreference resolution via hypergraph partitioning. In *Proceedings of the 23rd International Conference on Computational Linguistics (COLING-10)*, pages 143–151, 2010.

[5] A. K. Chandra, P. Raghavan, W. L. Ruzzo, R. Smolensky, and P. Tiwari. The electrical resistance of a graph captures its commute and cover times. *Computational Complexity*, 1996.

[6] A. E. Elo. *The Rating of Chess Players, Past and Present*. Arco Publishing, 1978.

[7] A. B. Goldberg, X. Zhu, and S. J. Wright. Dissimilarity in graph-based semi-supervised classification. In *International Conference on Artificial Intelligence and Statistics*, pages 155–162, 2007.

[8] R. Herbrich, T. Minka, and T. Graepel. TrueSkill$^{\text{TM}}$: A Bayesian Skill Rating System. In *Proceedings of the 20th conference Annual Conference on Neural Information Processing Systems (NIPS-06)*, pages 569–576, 2006.

[9] M. Herbster. Exploiting cluster-structure to predict the labeling of a graph. *Algorithmic Learning Theory*, 2008.

[10] Y. P. Hou. Bounds for the least Laplacian eigenvalue of a signed graph. *Acta Mathematica Sinica*, 21(4):955–960, 2005.

[11] S. Klamt, U.-U. Haus, and F. Theis. Hypergraphs and Cellular Networks. *PLoS Computational Biology*, 5(5), May 2009.

[12] D. J. Klein and M. Randić. Resistance distance. *Journal of Mathematical Chemistry*, 12(1):81–95, 1993.

[13] Y. Koren, L. Carmel, and D. Harel. ACE: a fast multiscale eigenvectors computation for drawing huge graphs. In *Information Visualization, 2002. INFOVIS 2002. IEEE Symposium on*, pages 137–144, 2002.

[14] J. Kunegis, S. Schmidt, A. Lommatzsch, J. Lerner, E. W. De Luca, and S. Albayrak. Spectral analysis of signed graphs for clustering, prediction and visualization. In *Proceedings of the 10th SIAM International Conference on Data Mining (SDM-10)*, volume 10, pages 559–559, 2010.

[15] T. Ricatte, R. Gilleron, and M. Tommasi. Hypernode Graphs for Spectral Learning on Binary Relations over Sets. In *Machine Learning and Knowledge Discovery in Databases (ECML PKDD 2014)*, pages 662–677. Springer, 2014.

[16] J. Rodríguez. On the Laplacian spectrum and walk-regular hypergraphs. *Linear and Multilinear Algebra*, 51(3):285–297, 2003.

[17] U. Von Luxburg. A tutorial on spectral clustering. *Statistics and computing*, 17(4):395–416, 2007.

[18] S. Zhang, G. D. Sullivan, and K. D. Baker. The automatic construction of a view-independent relational model for 3-D object recognition. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 15(6):531–544, 1993.

[19] D. Zhou, J. Huang, and B. Schölkopf. Learning from labeled and unlabeled data on a directed graph. In *Proceedings of the 22nd International conference on Machine learning (ICML-05)*, pages 1036–1043. ACM, 2005.

[20] D. Zhou, J. Huang, and B. Schölkopf. Learning with hypergraphs: Clustering, classification, and embedding. In *Proceedings of the 20th conference Annual Conference on Neural Information Processing Systems (NIPS-06)*, pages 1601–1608, Cambridge, MA, 2007. MIT Press.

[21] X. Zhu, Z. Ghahramani, J. Lafferty, et al. Semi-supervised learning using gaussian fields and harmonic functions. In *Proceedings of the 20th International conference on Machine learning (ICML-03)*, volume 3, pages 912–919, 2003.