Hypernode Graphs for Learning from Binary Relations between Groups in Networks

Thomas Ricatte, Rémi Gilleron, Marc Tommasi

Hypernode Graphs for Learning from Binary Relations between Groups in Networks

Thomas Ricatte
Inria
Lille, France
thomas@ricatte.fr

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

Marc Tommasi
Lille University
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. We introduced hypernode graphs in [15] as a formal model able to represent group interactions and able to infer individual properties as well. Spectral graph learning algorithms were extended to the case of hypernode graphs. As a proof-of-concept, we have shown how to model multiple players games with hypernode graphs and that spectral learning algorithms over hypernode graphs obtain competitive results with skill ratings specialized algorithms. In this paper, we explore theoretical issues for hypernode graphs. We show that hypernode graph kernels strictly generalize over graph kernels and hypergraph kernels. We show 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. Moreover, we also present some issues concerning random walks and the resistance distance for 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 the friendship relation between users, or for computer networks with the connection relation 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 in online games with teams of players. Group interactions are also observed when dealing with clusters in computer networks or communities in social networks. However a goal remains 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 model to represent group interactions and able to infer individual properties as well. We define 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. Moreover, every node participating in a hyperedge is given a non negative real-valued weight. Roughly speaking, a hypernode models a group, a hyperedge models a binary 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 presented in Figure 1. It is easy to show that there is a one

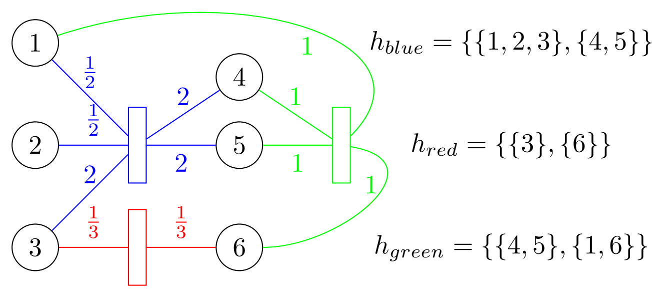


Figure 1: A hypernode graph with three hyperedges

to one correspondence between undirected graphs and hypernode graphs where all hypernodes are singleton sets if the weights satisfy an equilibrium condition.

Given relations between groups in a network, one of the goal is to evaluate individuals. We model the relations by a hypernode graph. The evaluation of individuals is done via real-valued node functions and the evaluation of groups is assumed to be a linear combination of the evaluation of the individuals in the group. As in the graph case, we assume a homophilic property at the group level : connected groups tend to be evaluated in a similar way. Thus, for every hyperedge, the weighted sum of node function values for two hypernodes are close enough. We introduce in [15] an unnormalized gradient G. We define an unnormalized Laplacian Δ for hypernode graphs by $\Delta = G^T G$, and the smoothness semi-norm Ω by, for every real-valued node function f, $\Omega(f) = f^T \Delta f$. And, we can show that $\Omega(f) = 0$ for constant node functions. Also, $\Omega(f)$ is close to 0 when the hyperedges satisfy the homophilic properties. Moreover, as the Laplacian Δ of every hypernode graph is positive semidefinite, we can use spectral graph learning algorithms [17], [19], [21] for learning in hypernode graphs.

As a proof-of-concept, we show how to use hypernode graphs and the associated spectral theory to infer skill ratings in the case of multiple players games. Given a set of games and the game outcomes, each player is represented by a node, each team of multiple players by a hypernode, and each game by a hyperedge between two teams with additional nodes modeling the game outcome. Thus, a skill rating function of players may be viewed as a real-valued node function over player nodes of the hypernode graph. And, if we denote by Δ the unnormalized Laplacian of the hypernode graph, we show that finding the optimal skill rating function reduces to finding the real-valued node function s^* minimizing $\Omega(s) = s^T \Delta s$ where s-values for the nodes modeling the game outcomes are fixed. Thus, we can apply semi-supervised learning algorithms. The inferred skill rating function allows to predict game outcomes for new games. We apply this learning method on real datasets of multiple players games to predict game outcomes in a semi-supervised, batch setting. Experimental results show that we obtain very competitive results compared to specialized algorithms such as Elo duelling [6] and TrueSkill [8].

One question raised by the introduction of hypernode graphs is whether hypernode graphs can be reduce to graphs. Indeed, one way is to consider hypergraphs where a hyperedge is a set of nodes. The notion of hypergraphs has recently received some attention in the machine learning literature and has found a number of applications [11, 18, 4]. But, hypergraphs essentially represent n-ary relations between nodes and fail to model relations between *sets of* nodes. Moreover, it has been proved in [1] that every hypergraph Laplacian introduced so far correspond to a graph Laplacian using an adequate graph construction.

In this paper we show that hypernode graphs strictly extends the expressive power of graph with respect to the class of Laplacians they define. We show that that the class of hypernode graph Laplacians contains the class of graph Laplacians and the class of graph kernels. No graph construction allows to define a graph and a set of smooth functions over the graph which coincides with the set of smooth functions over a given hypernode graph. As an intermediate result, we observe that a convex linear combination of graph kernels (used in multiple kernel graph kernel learning as in [2]) is a hypernode graph kernel.

Second, we show 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. This result exhibits an interesting class of signed graphs for which the Laplacian computed as $D - W$ is a positive semidefinite operator. We also obtain an alternative representation of such signed graphs together with an interpretation using hypernode graphs. As a consequence, homophilic relations between groups (hypernodes) lead to non homophilic relations between individuals (nodes).

Another issue is to define a distance between nodes in a hypernode graph from the hypernode graph kernel and to study whether such a distance can be interpreted in the hypernode graph. Indeed, it has been shown for graphs that 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 distance between nodes of a hypernode graph using the hypernode graph kernel as in the graph case. We define a diffusion operator for hypernode graphs and we show that the distance can be expressed from the differences of potentials. But, it should be noted that we leave open the question of finding an algorithmic definition of connected components in hypernode graphs. Also, the question of finding an interpretation of the distance in terms of random walks is left open because negative terms are involved in the expression.

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 $|H| = p$. Each hyperedge $h \in H$ is an unordered pair $\{s_h, t_h\}$ of two non empty and disjoint hypernodes (a hypernode is a subset of V). 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 $\sum_{i \in t_h} \sqrt{w_h(i)} = \sum_{i \in s_h} \sqrt{w_h(i)}$.

Note that 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. It should be noted that h_{red} can be viewed as an edge with edge weight $1/3$.

In order to implement the homophilic relation between hypernodes in hypernode graphs, we define the *unnormalized gradient* of a hypernode graph $\mathbf{h} = (V, H)$ 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. When the hypernode graph is a graph (all hypernodes are singleton sets), then the hypernode graph gradient is equal to the graph gradient. 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 $\mathbf{h}$. It should be noted that, as in the graph case, 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. We now show 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 $P = \{1, \dots, n\}$ 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 non negative

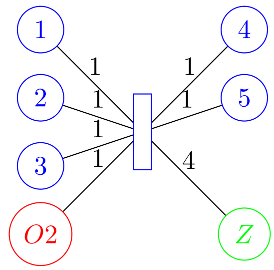


Figure 2: Hyperedge h for a game $A = \{1, 2, 3\}$ against $B = \{4, 5\}$ where B wins with score 2. Note the outcome node $O2$ corresponding to the score 2 and the lazy node Z. All weights are set to 1 except the weight for Z chosen in order to ensure the Equilibrium condition.

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 contribution or to be 1 without known 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. The construction is illustrated in Figure 2. 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 draw.

Let us illustrate how the hypernode graph allows to model player skills. For this, let us consider the game and the hypernode graph of Figure 2. 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 assuming an additive model for player skills.

Given the hypernode graph $\mathbf{h}$ constructed as above with Laplacian Δ, we show that the *skill rating problem* is equivalent to find the real-valued node function s solving the optimization problem

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

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 $\Delta_\mu^\dagger$.

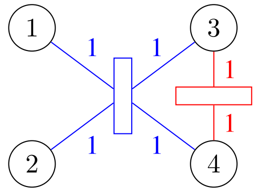
We have applied these two algorithms for the skill rating problem over simple tennis games, double tennis games and online games (Halo2 dataset). Experimental results show that we obtain very competitive results, in a batch setting, compared with specialized algorithms such as Elo duelling [6] and TrueSkill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

3 Hypernode Graphs, Graphs and Signed Graphs

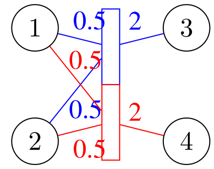
3.1 The Class of Hypernode Graphs Laplacians

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

- is closed by convex linear combination,



$$G = \begin{pmatrix} 1 & 1 & -1 & -1 \\ 0 & 0 & -1 & 1 \end{pmatrix}$$



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

Figure 3: two L-equivalent hypernode graphs with their respective gradient which are square root of the Laplacian Δ shown in Figure 4.

- is closed by pseudoinverse,
- and strictly contains the class of graph Laplacians and the class of graph kernels.

Linear combinations of graph kernels have been used for semi-supervised learning as in [2], it is worth noticing that a convex linear combination of graph kernels is a hypernode graph kernel (also a hypernode graph Laplacian). We propose the following

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 analytical form.

3.2 L-equivalent Hypernode Graphs

In the proof of Proposition 1, given a symmetric positive semidefinite real-valued matrix M such that $\mathbf{1} \in \mathrm{Null}(M)$, we 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 of the condition $\mathbf{1} \in \mathrm{Null}(M)$. This leads to a hypernode graph $\mathbf{h}$ for which it is immediate that its Laplacian is M. 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 3. It should be noted that there are hypernode graphs L-equivalent to a graph. We now present a construction of a L-equivalent –possibly signed– graph given a hypernode graph.

3.3 Pairwise Weight Matrix

Let $\mathbf{h}$ be a hypernode graph, we define for every node pair (i, j) and every hyperedge h, the hyperedge pairwise weight as $w_h(i, j) = 0$ when $i = j$, and for $i \neq j$,

$$w_h(i, j) = P(h, i, j)\sqrt{w_h(i)}\sqrt{w_h(j)} \ ,$$

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. The *pairwise weight matrix* W of a hypernode graph $\mathbf{h} = (N, H)$ is defined by

$$\forall i, j \in N, W_{i,j} = \sum_{h \in H} w_h(i, j) \ . \tag{2}$$

It can be shown that the pairwise weight matrix of a hypernode graph which is a graph is the adjacency matrix of the graph. Also, it is easy to show that, for every $i \neq j$ in V, we have $W_{i,j} = -\Delta_{i,j}$. It can also be shown that, for every i in V, if the degree of a node i is defined by $dg(i) = \sum_h \sqrt{w_h(i)}$, then we have $\sum_j W_{i,j} = dg(i)$. These two results allow to state that

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

Figure 4: Laplacian matrix, pairwise weight matrix, and degree matrix for the two L-equivalent hypernode graphs shown in Figure 3.

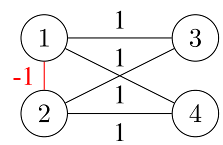


Figure 5: The reduced signed graph with adjacency matrix W of Figure 4 of the two L-equivalent hypernode graphs shown in Figure 3.

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 degree matrix of* $\mathbf{h}$*. Then, the Laplacian of* $\mathbf{h}$ *is* $\Delta = D - W$.

An example is shown in Figure 4. 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.*

3.4 Hypernode Graphs and Signed Graphs

Let us consider the Laplacian matrix Δ shown in Figure 4 of the two L-equivalent hypernode graphs shown in Figure 3. The matrix Δ is equal to $D - W$ and it can be noted that the pairwise weight matrix W contains negative weights. The matrix W can be seen as the adjacency matrix of a signed graph shown in Figure 5. And, we define the *reduced signed graph* of a hypernode graph $\mathbf{h}$ to be the signed graph g with adjacency matrix W, where W is the pairwise weight matrix of the hypernode graph $\mathbf{h}$. 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 L-equivalent to a hypernode graph if and only if the matrix $D - W$ is positive semidefinite.

It should be noted that, for signed graphs, the matrix $D - W$ can be indefinite. Thus, many works have studied the notion of Laplacian for signed graphs. Koren et al in [13] have considered the class of signed graphs such that $D - W$ is positive semidefinite, i.e. the class of signed graphs L-equivalent to hypernode graphs. Herbster in [9] propose to consider the class of symmetric diagonally dominant matrices with nonnegative diagonal entries. While others ([10, 7, 14]) have used a modified definition of the Laplacian to obtain a positive semidefinite Laplacian.

3.5 Hypernode Graph Laplacians can not be reduced to Graph Laplacians

We have already shown that the class of hypernode graph Laplacians strictly contains the class of graph Laplacians and the class of graph kernels. But it could be the case that the smoothness measure Ω defined by a hypernode graph laplacian can be defined via a graph Laplacian for an adequate graph construction. 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 an adequate graph construction. One important graph construction is the clique expansion where each hyperedge is

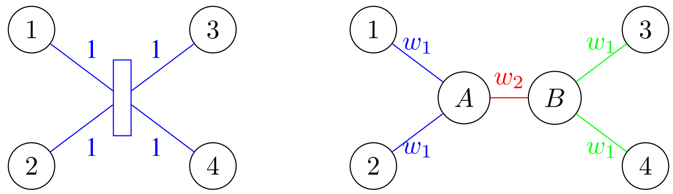


Figure 6: A hypernode graph and a candidate star expansion but no expansion exists that defines the same smooth functions.

replaced by a clique graph with uniform weights. A second graph construction is the star expansion where, for every hyperedge, a new node is added and is linked with all the nodes in the hyperedge.

While we can think of similar constructions for the case of hypernode graphs, we have proved 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 6. We show by contradiction that there does not exists a finite graph whose node set contains $\{1, 2, 3, 4\}$ which can express that $(f(1) + f(2) - f(3) - f(4))^2 = 0$.

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.

4.1 Defining a distance in Hypernode Graphs

Let us consider throughout the section a hypernode graph $\mathbf{h} = (N, H)$. We denote by Δ its Laplacian matrix and we consider the hypernode graph kernel to be the Moore-Penrose pseudo-inverse $\Delta^\dagger$. Let us define d by, for every i, j in N,

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

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: $d(i,j) = 0 \Rightarrow i = j$, which is not true in general. Indeed, let us consider the hypernode graph in Figure 6, we have $d(1,2) = 0$. 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 $j \in N$ called *sink* node, we consider the input function $\mathbf{In}_j$ defined by $\mathbf{In}_j(j) = dg(j) - \mathrm{Vol}(\mathbf{h})$ and $\mathbf{In}_j(i) = dg(i)$ 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

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

where $\mathbf{e}_i$ is the unit vector with 1 in component i. We are now in order to 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* $i \neq j$,

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

We omit the proof by lack of space.

4.3 Connected Components and Random Walks in Hypernode Graphs

We have seen that the condition for d defined in Equation 3 to be a metric is $\mathrm{Null}(\Delta) = \mathrm{Span}(\mathbf{1})$. By analogy with the case of graphs, this condition can be viewed as a definition of a connected hypergraph. The intuition is that, for a connected hypernode graph, nodes can not be labeled independently. We have defined connected components on the reduced signed graphs, but, so far, we have not found an algorithmic definition of connected components in hypernode graphs.

Another isue 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

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

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). And, as a particular case, we get that 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 in the case of graphs [12, 5].

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

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

The term $p(h|i)$ can be interpreted as a jumping probability from 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 simple interpretation of the distance d^2 in terms of random walks in the hypernode graph.

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 addtive model for indivual rates. We can also model dominace 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, 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$^{\mathrm{TM}}$: 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.