Sample Complexity of Learning Multi-Value Opinions in Social Networks

Masato Shinoda[1], Yuko Sakurai[2], Satoshi Oyama[3]

[1] Nara Women's University
shinoda@cc.nara-wu.ac.jp
[2] Nagoya Institute of Technology
sakurai@nitech.ac.jp
[3] Hokkaido University
oyama@ist.hokudai.ac.jp

Abstract. We consider how many users we need to query in order to estimate the extent to which multi-value opinions (information) have propagated in a social network. For example, if the launch date of a new product has changed many times, the company might want to know to which people the most current information has reached. In the propagation model we consider, the social network is represented as a directed graph, and an agent (node) updates its state if it receives a stronger opinion (updated information) and then forwards the opinion in accordance with the direction of its edges. Previous work evaluated opinion propagation in a social network by using the probably approximately correct (PAC) learning framework and considered only binary opinions. In general, PAC learnability, i.e., the finiteness of the number of samples needed, is not guaranteed when generalizing from a binary-value model to a multi-value model. We show that the PAC learnability of multi-value opinions propagating in a social network. We first prove that the number of samples needed in a multi-opinion model is sufficient for $(k-1)\log(k-1)$ times the number of samples needed in a binary-opinion model, when k (≥ 3) is the number of opinions. We next prove that the upper and lower bounds on the number of samples needed to learn a multi-opinion model can be determined from the Natarajan dimension, which is a generalization of the Vapnik-Chervonenkis dimension.

Keywords: PAC Learning · Opinion Estimation · Social Network

1 Introduction

Today, many people exchange various opinions and information via social network services (SNSs). Therefore, SNSs have become an important means for individuals, companies, and other information sources to disseminate and propagate information such as opinions and product advertisements. To measure the effectiveness of an advertising campaign, it is necessary to know to whom the information has reached. However, for a large network, it is impractical to check

whether everyone has received the information, so we have to rely on estimation based on sampling.

A recently proposed method [4] for estimating the extent of propagation of binary opinions in a social network uses the probably approximately correct (PAC) learning framework [10] to obtain the number of samples needed for estimation. The PAC learning framework is typically used for estimating the number of samples needed to learn an accurate classification model with high probability. PAC learning is a classical theory in the field of machine learning that has in recent years been applied to various multi-agent system (MAS) problems. For example, PAC learning has been applied to cooperative games [1, 7, 9, 11] and has been combined with incentive design [12].

Conitzer et al. represented a social network as a directed graph with nodes as agents and edges as relations among agents and assumed that opinions propagated in accordance with the direction of edges [4]. The state of each agent is binary: either it has received an opinion (1) or it has not (0). Once an opinion enters an agent node through an incoming edge, the agent propagates the opinion through its outgoing edges. PAC learning is used to calculate the order of the number of samples (agents) that need to be asked whether they have an opinion needed to estimate, within a predefined error margin, the opinions of other agents in the network once opinion propagation has completed.

In their study, Conitzer et al. considered only binary opinions, but opinions are not always expressed in binary form. For example, a company may announce the scheduled release date of a new product and then change the date or product specifications. In this case, the company would probably want to know the extent to which the old information and the updated information have propagated. As another example, knowledge about a new infectious disease is constantly being updated, and knowing who has each stage of the information would be helpful in implementing effective countermeasures. Thus, the degree to which opinions and information have spread in social networks can have multiple uses. It is also practically necessary to determine the sample size needed to estimate the extent to which opinions have been reached in a social network.

Here, we generalize the problem setting used by Conitzer et al. [4]. More specifically, when $k(\geq 3)$ types of opinions (labels) propagate in a social network represented by a directed graph, we use PAC learning to determine the number of samples required to estimate labels other agents have when opinion propagation is completed. In general, when a binary model is extended to a multi-valued model, PAC learnability, i.e., the finiteness of the required number of samples, is not guaranteed. However, in our model of opinion propagation on a directed graph, generalization is possible because the hypothesis class $\mathcal{H}$ (the set of candidate labelings to be estimated) representing possible labeling patterns is determined from the propagation conditions of labels on the graph. We first show, as a relative comparison, that for a k (≥ 3)-valued model, $(k-1)\log(k-1)$ times the number of samples required for PAC learning of a binary model is sufficient. Next, we determine specific upper and lower bounds on the number of samples required for PAC training of a k (≥ 3)-valued model.

The rest of this paper is organized as follows. We first introduce previous related work in Section 2. In Section 3, we describe PAC learning for binary and multi-valued models. In Section 4, we present the problem setting considered in this paper. We compare binary and multi-valued models in Section 5. In Section 6, we define the Natarajan dimension in our problem setting and determine the upper and lower bounds on the number of samples needed for estimating the overall propagation of multiple opinions in a social network.

2 Previous Work

As mentioned in Section 1, PAC learning has been applied to various MAS problems.

Procaccia and Rosenschein investigated PAC learning for simple cooperative games in which the coalitions are partitioned into winning and losing coalitions. They analyzed the complexity of learning a suitable concept class via the Vapnik-Chervonenkis (VC) dimension and developed an algorithm that learns its class [9]. Balcan et al. concentrated on the core stability in cooperative games when agents' preferences are fully unknown [1]. They established a connection between PAC learnability and core stability: for games that are efficiently learnable, payoff divisions that are likely to be stable can be found by using a polynomial number of samples. Their defining of PAC stability led to studies of PAC stability. Sliwinski and Zick studied PAC stability in hedonic games (which are a variant of cooperative games) when agents' preferences are fully unknown [11].

Berenbrink et al. investigated an agent-based model for opinion formation in a social network in which the opinion of an agent depends both on its own intrinsic opinion and on those of its network neighbors [3]. They analyzed the convergence time of asynchronous Hegselmann-Krause opinion dynamics in arbitrary social networks. Irfan et al. addressed the problem of how to maximize the spread of information while minimizing the spread to unintended recipients under budget constraints [8].

Zhang and Conitzer considered PAC learning in the presence of strategic manipulation [12]. They addressed an incentive problem in which a point being classified may strategically modify its features in order to receive a more desirable outcome. Incentive design is an important MAS research area, and Zhang and Conitzer presented a new direction for combining machine learning and MAS techniques.

3 PAC Learning

In this section, we present several fundamental concepts of PAC learning.

3.1 PAC learning for binary labels

PAC learning provides an estimate of the number of samples required to learn a hypothesis that guarantees a confidence level δ and generalization error ε for

a learning problem with a hypothesis class $\mathcal{H}$. Suppose there exists a data set X ($|X| = n$) and a label set $W = \{0, 1, 2, \ldots, k-1\}$ ($k \geq 2$). Suppose that each data item $x \in X$ is given a label $f(x) \in W$, and let W^X be the set of all possible labelings on X. The learner independently selects m points on the basis of probability distribution $\mathcal{D}$ on X and knows the value of f at each point. As an estimation of the label value $f = \{f(x)\}_{x \in X}$ from the sample data, one h is selected from the predefined hypothesis set $\mathcal{H} \subseteq W^X$.

The error between the true labeling f and its hypothesis h is defined as the probability that $f(x)$ and $h(x)$ differ when x is chosen on the basis of $\mathcal{D}$ and is denoted as

$$L_{\mathcal{D},f}(h) = P_{\mathcal{D}}(f(x) \neq h(x)), \tag{1}$$

where $P_{\mathcal{D}}$ denotes the probability of choosing the point $x \in X$ in accordance with $\mathcal{D}$. That is, $L_{\mathcal{D},f}(h)$ represents the error rate (measured in $\mathcal{D}$) of h when f is the correct answer.

For a given $\varepsilon, \delta > 0$, the goal of PAC learning is to determine the hypothesis $h \in \mathcal{H}$ satisfying $L_{\mathcal{D},f}(h) \leq \varepsilon$ with probability greater than $1 - \delta$. The procedure to achieve this goal is outlined below.

1. The learner is given a hypothesis class $\mathcal{H}$, which is a subset of all possible labelings W^X. The learner knows nothing about the true labeling f to be guessed except that it is an element of W^X.
2. The m points $(x_1, x_2, \ldots, x_m) \in X^m$ chosen from X and the values of f at those points $(f(x_1), f(x_2), \ldots, f(x_m))$ are called a training set. The learner determines an algorithm to find a hypothesis $h \in \mathcal{H}$ as a guess of f from the given training set. We call the algorithm A henceforth.
3. True labeling $f \in W^X$ and probability distribution $\mathcal{D}$ on X are determined.
4. The m points $(x_1, x_2, \ldots, x_m)$ chosen independently from X in accordance with probability distribution $\mathcal{D}$ and labels $(f(x_1), f(x_2), \ldots, f(x_m))$ at those points are given to the learner as a training set. The learner determines hypothesis $h \in \mathcal{H}$ using algorithm A.
5. The obtained hypothesis h is evaluated using $L_{\mathcal{D},f}(h)$, and if the value obtained is less than ε, the learning is considered successful.

The m points $x_1, x_2, \ldots, x_m$ chosen in step 2 must be considered in the case of overlap. The learner must decide algorithm A before step 3, i.e., without any information about $f, \mathcal{D}$. That is, A is a mapping from $(X \times W)^m$ to $\mathcal{H}$. Let $m_L(\varepsilon, \delta)$ be the smallest integer m for which algorithm A can be defined such that the probability of success in guessing is uniformly greater than $1 - \delta$ regardless of $\mathcal{D}$ and f. The condition corresponding to the assumption that f given in step 3 is in particular an element of $\mathcal{H}$; i.e.

$$\min_{h \in \mathcal{H}} L_{\mathcal{D},f}(h) = 0 \tag{2}$$

is called the realizability condition. Here, PAC learning is performed under the assumption of this condition.

3.2 Vapnik-Chervonenkis dimension

The VC dimension is used to characterize the sample complexity of learning a given hypothesis class $\mathcal{H}$ for a binary label set $W = \{0, 1\}$.

Definition 1. VC dimension *Let X be a non-empty set, and let $\mathcal{H}$ be a hypothesis class of functions from X to $\{0, 1\}$; i.e., $\mathcal{H} \subseteq \{0, 1\}^X$. We say that a subset $S \subseteq X$ is* ***shattered*** *by $\mathcal{H}$ if the restriction $\mathcal{H}|_S$ of $\mathcal{H}$ to S is identical to $\{0, 1\}^S$. The VC dimension of a hypothesis class $\mathcal{H}$, denoted $d_{VC}(\mathcal{H})$, is the maximum size of a set $S \subseteq X$ that can be shattered by $\mathcal{H}$. If $\mathcal{H}$ can shatter sets of arbitrarily large size, we say that $\mathcal{H}$ has an infinite VC dimension.*

In other words, the VC dimension is the maximum size of a set of agents in which all combinations for any labels are feasible.

Example 1. We set $W = \{0, 1\}$ and $X = \{a, b, c\}$. If we denote the elements of $\mathcal{H}$ in the form $(h(a), h(b), h(c))$, the hypothesis class is $\mathcal{H} = \{(1, 0, 0), (0, 1, 0), (0, 0, 1), (1, 1, 0), (1, 0, 1), (0, 1, 1)\}$.

If we set $S_{abc} = \{a, b, c\}$, we have $\mathcal{H}|_{S_{abc}} = \mathcal{H}$. $\mathcal{H}|_{S_{abc}}$ does not include $(0, 0, 0)$ and $(1, 1, 1)$, so S_{abc} is not shattered by $\mathcal{H}$. If we set $S_{ab} = \{a, b\}$, we have $\mathcal{H}|_{S_{ab}} = \{(0, 0), (0, 1), (1, 1), (1, 0)\}$. $\mathcal{H}|_{S_{12}}$ consists of all feasible combinations for W. As a result, S_{ab} is shattered by $\mathcal{H}$, and $d_{VC}(\mathcal{H}) = 2$. Furthermore, if we set $S_{ac} = \{a, c\}$ and $S_{bc} = \{b, c\}$, both S_{ac} and S_{bc} are shattered by $\mathcal{H}$.

We estimate $m_L(\varepsilon, \delta)$ by applying the VC dimension under realizability assumption (2).

Theorem 1. [10] *Let $\mathcal{H}$ be a hypothesis class of functions from a domain X to $\{0, 1\}$; that is, $\mathcal{H} \subseteq \{0, 1\}^X$. Assume that $d_{VC}(\mathcal{H}) < \infty$. Then, there are absolute constants $C_1, C_2 > 0$ such that*

$$C_1 \frac{d_{VC}(\mathcal{H}) + \log(1/\delta)}{\varepsilon} \leq m_L(\varepsilon, \delta) \leq C_2 \frac{d_{VC}(\mathcal{H}) \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}. \quad (3)$$

This fundamental theorem states that, even if X is an infinite set, $\mathcal{H}$ is PAC learnable if its VC dimension is finite.

3.3 PAC learning for multiple labels

We introduce PAC learning for $k(\geq 2)$ labels $W = \{0, 1, 2, \ldots, k-1\}$ under realizability assumption (2) in which there exists a true label f in hypothesis class $\mathcal{H}$.

To estimate the accuracy of hypothesis h, we define $l_{\mathcal{D},f}(h)$ in addition to error rate $L_{\mathcal{D},f}(h)$:

$$l_{\mathcal{D},f}(h) = E_{\mathcal{D}}(|f(x) - h(x)|). \quad (4)$$

$l_{\mathcal{D},f}(h)$ represents the expected value of the error rate, taking into account the difference in size between f and h, measured with respect to probability distribution $\mathcal{D}$. From this definition, we get

$$L_{\mathcal{D},f}(h) \leq l_{\mathcal{D},f}(h) \leq (k-1) L_{\mathcal{D},f}(h). \quad (5)$$

(a) Hypothesis H

a	0	0	1	1
b	0	2	0	2
c	0	2	0	2

(b) Function g

	a	b	c
0	0	2	2
1	1,2	0,1	0,1

(c) Labels g

a	0	0	1	1
b	1	0	1	0
c	1	0	1	0

Fig. 1. Example of calculating Graph dimension

If $k = 2$, we have $L_{\mathcal{D},f}(h) = l_{\mathcal{D},f}(h)$ because the only possible values of f and h are 0 and 1. In PAC learning for multiple labels, when hypothesis h is estimated using $l_{\mathcal{D},f}(h)$ or $L_{\mathcal{D},f}(h)$, we say that the learner has succeeded in its prediction if the estimation does not exceed ε. Furthermore, the amount of training data needed so that the probability of successful prediction is equal to or more than $1 - \delta$ is denoted by $m_L(\varepsilon, \delta)$ or $m_l(\varepsilon, \delta)$ in accordance with the evaluation criterion.

To extend the VC dimension so that we can measure the complexity of hypothesis class $\mathcal{H}$ in the multi-valued case, consider $\mathcal{H}|_S$, which restricts $\mathcal{H}$ to S. As in Definition 1, one might think that one should look for the largest S such that $\mathcal{H}|_S$ coincides with $\{0, 1, \ldots, k-1\}^S$ and define $d_{VC}^{(k)}(\mathcal{H})$ as the number of its elements. However, it is known that, for $k > 2$, the case $m_L(\varepsilon, \delta), m_l(\varepsilon, \delta) = \infty$ occurs even if $d_{VC}^{(k)}(\mathcal{H}) < \infty$. Therefore, $d_{VC}^{(k)}(\mathcal{H})$ defined above is not useful for determining PAC learnability in the multi-valued case.

3.4 Graph and Natarajan dimensions

We give the definitions of the Graph and Natarajan dimensions for PAC learning for multiple labels [5].

Definition 2. (Graph dimension) *Let X be a non-empty set, and let $\mathcal{H}$ be a hypothesis class of functions from X to $\{0, 1, \ldots, k-1\}$; i.e., $\mathcal{H} \subseteq \{0, 1, \ldots, k-1\}^X$. We say that a subset $S \subseteq X$ is **G-shattered** by $\mathcal{H}$ if there exists a $\{0, 1, 2, \ldots, k-1\}$-value function g over S such that, for any subset $T \subseteq S$, there exists an $h \in \mathcal{H}$ such that*

$$\forall x \in T,\ h(x) = g(x), \text{and } \forall x \in S \backslash T,\ h(x) \neq g(x). \tag{6}$$

The Graph dimension of hypothesis class $\mathcal{H}$, denoted $d_G(\mathcal{H})$, is the maximum size of a set $S \subseteq X$ that can be G-shattered by $\mathcal{H}$.

Function g converts multiple labels into binary labels by separating k labels into a single label and a set containing the remaining labels. It may differ by agent. The Graph dimension is the maximum size of a set of agents in which certain combinations of labels determined in accordance with g are feasible. We show an example of determining the Graph dimension.

(a) Hypothesis H

a	0	0	1	1
b	0	2	0	2
c	0	2	0	2

(b) Functions g_1, g_2

	a	b	c
g_1	0	2	0
g_2	1	0	2

(c) Labels g_1, g_2

a	g_1	g_1	g_2	g_2
b	g_2	g_1	g_2	g_1
c	g_1	g_2	g_1	g_2

Fig. 2. Example of calculating Natarajan dimension

Example 2. Let W be $\{0, 1, 2\}$ and X be $\{a, b, c\}$. If we denote the elements of $\mathcal{H}$ in the form $(h(a), h(b), h(c))$, the hypothesis class is $\mathcal{H} = \{(0, 0, 0), (0, 2, 2), (1, 0, 0), (1, 2, 2)\}$, as shown in table (a) in Figure 1.

Let us set function g as shown in table (b) in Figure 1. Function g relabels 0 as 0 and 1 and 2 as 1 for agent a and relabels 2 as 0 and 0 and 1 as 1 for agents b and c. There exist all combinations of labels determined in accordance with g in $S = \{a, b\}$ shown in table (c) in Figure 1. Furthermore, setting $S = \{a, c\}$ creates a set of agents in which all combinations for labels determined in accordance with function g are feasible. If $S = \{a, b, c\}$, certain combinations for labels determined in accordance with functions g are infeasible. As a result, the Graph dimension is 2.

Definition 3. (Natarajan dimension) *Let X be a non-empty set, and let $\mathcal{H}$ be a hypothesis class of functions from X to $\{0, 1, \ldots, k-1\}$; i.e., $\mathcal{H} \subseteq \{0, 1, \ldots, k-1\}^X$. We say that a subset $S \subseteq X$ is **N-shattered** by $\mathcal{H}$ if there exists $\{0, 1, 2, \ldots, k-1\}$-value functions g_1 and g_2 over S such that for any subset $T \subseteq S$, there exists an $h \in \mathcal{H}$ such that*

$$\forall x \in T,\ h(x) = g_1(x), \text{and } \forall x \in S \backslash T,\ h(x) = g_2(x). \tag{7}$$

The Natarajan dimension of hypothesis class $\mathcal{H}$, denoted $d_N(\mathcal{H})$, is the maximum size of a set $S \subseteq X$ that can be N-shattered by $\mathcal{H}$.

Intuitively, we first select binary labels from multiple labels. Then we consider the selected binary labels as g_1 and g_2; g_1 and g_2 may differ by agent. The Natarajan dimension is the maximum size of a set of agents in which all combinations for labels determined in accordance with functions g_1 and g_2 are feasible.

Example 3. Let W be $\{0, 1, 2\}$ and X be $\{a, b, c\}$. If we denote the elements of $\mathcal{H}$ in the form $(h(a), h(b), h(c))$, the hypothesis class is $\mathcal{H} = \{(0, 0, 0), (0, 2, 2), (1, 0, 0), (1, 2, 2)\}$, as shown in table (a) in Figure 2.

Let us set functions g_1 and g_2 as shown in table (b) in Figure 2. For agent a, 0 and 1 indicate g_1 and g_2, respectively. There exist all combinations of labels determined in accordance with g_1 and g_2 in $S = \{a, b\}$, as shown in table (c) in Figure 2. Furthermore, setting $S = \{a, c\}$ creates a set of agents in which all combinations for labels determined in accordance with functions g_1 and g_2 are feasible. If $S = \{a, b, c\}$, all combinations for labels determined in accordance

with functions g_1 and g_2 are infeasible. As a result, the Natarajan dimension is 2.

The Natarajan and Graph dimensions are identical to the VC dimension when $W = \{0, 1\}$. From the definitions, if $d_{VC}^{(k)}(\mathcal{H}) \leq d_N(\mathcal{H}) \leq d_G(\mathcal{H})$ for any $k(\geq 2)$, we can estimate $m_L(\varepsilon, \delta)$ by using the Natarajan and Graph dimensions as follows.

Theorem 2. ([5]) *Let us assume that $d_G(\mathcal{H}) < \infty$. For the constants C_1, C_2, from Theorem 1, for every $\mathcal{H}$, we have*

$$C_1 \frac{d_N(\mathcal{H}) + \log(1/\delta)}{\varepsilon} \leq m_L(\varepsilon, \delta) \leq C_2 \frac{d_G(\mathcal{H}) \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon}. \tag{8}$$

The Natarajan and Graph dimensions can be used to estimate the amount of training data required for a general multi-valued model. Furthermore, Ben-David et al. proved the following relationships between $d_N(\mathcal{H})$ and $d_G(\mathcal{H})$ [2].

$$d_N(\mathcal{H}) \leq d_G(\mathcal{H}) \leq 4.67 \log_2 k \; d_N(\mathcal{H}). \tag{9}$$

Thus, there is an equivalence between $d_N(\mathcal{H})$ being finite and $d_G(\mathcal{H})$ being finite.

4 Problem Setting

Let $G = (V, E)$ be a directed graph. The node set V corresponds to the set X in the general theory described in the previous section. When V is a finite set, the number of nodes is $|V| = n$. Let $(u, v) \in E$ denote the directed edge from node u to node v.

If there exists a sequence of nodes $u, u_1, u_2, \ldots, u_{l-1}, v$ connecting two nodes $u, v \in V$ by directed edges ($u_0 = u, u_l = v$, then, for $1 \leq i \leq l$ $(u_{i-1}, u_i) \in E$), the relationship is denoted as $u \to v$. Then, let $\rho(u, v)$ be the number of edges (l in the above notation) when the two nodes are connected by the shortest directed edge sequence. If $u \to v$ does not hold, then $\rho(u, v) = \infty$. Note that we write $\rho_U(u, v)$ when we restrict the sequence of nodes connecting two nodes to those contained in the subset U of V. For $u, v \in U$, $\rho(u, v) \leq \rho_U(u, v)$.

The set of labels given to each node is $W = \{0, 1, 2, \ldots, k-1\}$. The case $k = 2$ corresponds to binary labels. Each node u is given one of these k kinds of values, which we denote by $f(u)$. A greater or lesser value of this label represents greater or lesser information about a certain matter, and this information spreads through the directed edges of the graph; i.e.,

$$u \to v \quad \Longrightarrow \quad f(u) \leq f(v) \tag{10}$$

is assumed to be satisfied.

In Conitzer et al.'s model [4], the state of each node is binary ($k = 2$) (it has or does not have the target information), and the set of nodes $S_0 \subseteq V$ that initially have that information is given as the initial state. The final state is represented

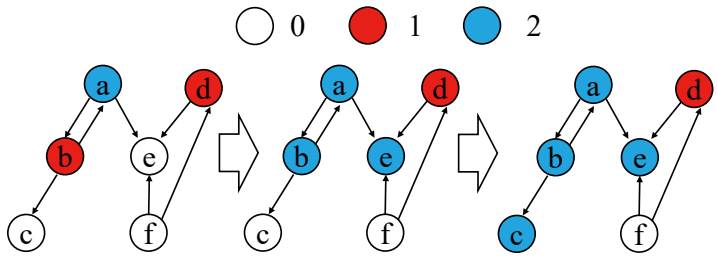


Fig. 3. Example of opinion flow

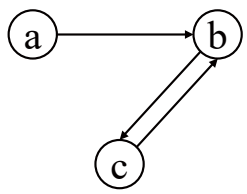


Fig. 4. Example of social network

by S_∞, and the objective is to estimate it. Once the information is acquired, it is not forgotten; i.e., if $u \in S_0$, then $u \in S_\infty$ is assumed. It is also assumed that the information is transmitted through directed edges; i.e., if $u \in S_0$ and $u \to v$, then $v \in S_\infty$ is also assumed. In our model, the above condition (10) is given by specifying the conditions that the final state must satisfy. We assume that each node u obtains information from nodes other than those connected by directed edges, and we allow the states $u \to v$ and $f(u) < f(v)$.

Example 4. Figure 3 shows how multiple opinions propagate in a social network. In this example, three opinions propagate. Initially, agent a has 2, agents b and d have 1, and the other agents have 0 as an opinion. Then, they share their opinions in accordance with the directions of their outgoing edges. Next, agents a, b, and e have 2 and d has 1. The opinion of agent b changes from 1 to 2 because 2 is stronger than 1. Although agent e receives both 1 and 2, opinion 2 is kept for the same reason. Finally, agents a, b, c, and e have 2, agent d has 1, and agent f still has 0.

Of all possible labeling sets W^V on V, the subset that satisfies condition (10) is the hypothesis class and is denoted by $\mathcal{H}$. Hypothesis class $\mathcal{H}$, which is restricted to a subset V' of the node set V, is denoted by $\mathcal{H}|_{V'}$.

Example 5. Let $V = \{a, b, c\}$ denote the node set $E = \{(a, b), (b, c), (c, b)\}$, the edge set, as shown in Figure 4. Let $k = 3$ be the number of labels. If we denote

the elements of $\mathcal{H}$ in the form $(h(a), h(b), h(c))$, the hypothesis class is

$$\mathcal{H} = \{(0,0,0), (0,1,1), (0,2,2), (1,1,1), (1,2,2), (2,2,2)\};$$

and if we restrict the node set to $V' = \{a, c\}$, the restricted hypothesis class is

$$\mathcal{H}|_{V'} = \{(0,0), (0,1), (0,2), (1,1), (1,2), (2,2)\}.$$

In Conitzer et al. ' s model [4] and the model used in this study, the learner is given information on the directed graph $G = (V, E)$ in step 1 of the learning procedure, as described in Section 3.1. At this point, the learner can define a subset of W^V that satisfies condition (10) as the hypothesis class $\mathcal{H}$. In the case of binary labels treated in Conitzer et al. ' s study, the VC dimension of the hypothesis class $\mathcal{H}$ satisfying condition (10) is determined to be the maximum number of nodes in the graph that do not affect each other. Since the value of this dimension is determined from G, it is written as $d_{VC}(G)$. That is, the following holds for binary labels.

$$d_{VC}(G) = \max_{U \subseteq V}\{\ |U| \mid {}^{\forall}u, v \in U, \rho_U(u, v) = \infty\} \tag{11}$$

Note that, if there exists a cycle in a directed graph G, i.e., a sequence of directed edges starting from $u \in V$ and returning to u, the label values of the nodes in the cycle must all be identical by condition (10). Therefore, it is sufficient to consider a directed acyclic graph in which all the nodes in a cycle in G are reduced to a single node, thereby removing all cycles. A directed acyclic graph can also be regarded as a partially ordered set. A subset of a partially ordered set for which none of its elements can be ordered is called an antichain. The number of elements in the largest antichain is called the width of the partially ordered set, which is in fact the right-hand side of Equation (11). It is known that the width of a partially ordered set can be determined by the polynomial order of the number of elements, which is the number of nodes in the corresponding directed acyclic graph [6]. Thus, variable d in inequality (3) of Theorem 1, which represents the estimated amount of training data needed, can be replaced by the width of the partially ordered set, which is a directed acyclic graph with cycles removed from the directed graph G.

In the following, we consider the case in which there are k kinds of labels $W = \{0, 1, 2, \ldots, k-1\}$, and hypothesis class $\mathcal{H}$, which is a set of candidate labelings of the entire V, is defined as the set of all f satisfying condition (10). In other words, realizability condition (2) is necessarily satisfied here. The learning problem is therefore to find f among the elements of the set $\mathcal{H}$ of W-valued functions on V. If parameters ε and δ are given, the amounts of training data required for PAC learning on $l_{\mathcal{D},f}(h)$ and $L_{\mathcal{D},f}(h)$ are defined as $m_l^{(k)}(\varepsilon, \delta)$ and $m_L^{(k)}(\varepsilon, \delta)$, respectively.

5 Comparison between Binary and Multi-valued Models

In this section, we evaluate $m_l^{(k)}(\varepsilon, \delta)$ for a k-valued model compared with the binary model by using the following theorem.

Theorem 3. *Let the number of labels be $k \geq 2$. The following holds for $m_l^{(k)}(\varepsilon, \delta)$, i.e., the amount of training data required for PAC labeling.*

$$m_l^{(2)}\left(\frac{\varepsilon}{k-1}, \delta\right) \leq m_l^{(k)}(\varepsilon, \delta) \leq m_l^{(2)}\left(\frac{\varepsilon}{k-1}, \frac{\delta}{k-1}\right) \tag{12}$$

This theorem says that approximately $(k-1)\log(k-1)$ times the number of samples is sufficient for PAC learning for the k-valued model compared with the binary model. As for $m_l^{(k)}(\varepsilon, \delta)$ regarding the error rate, a similar evaluation formula can be obtained from the theorem and inequality (5).

Proof. The inequality on the left-hand side in Theorem 3 is shown because the error rate is exactly $k-1$ times the error rate when the case in which the label of each node takes the value $\{0, 1\}$ in the binary model matches the case in which the label takes the value $\{0, k-1\}$ in the k-valued model. The right-hand side inequality is shown below. Since the label of each node $v \in V$ can take k different values of $W = \{0, 1, 2, \ldots, k-1\}$, we divide these values into the following $k-1$ binary models: (i) $f(v) = 0$ or $f(v) > 0$, (ii) $f(v) \leq 1$ or $f(v) > 1$, and (iii) $f(v) \leq 2$ or $f(v) > 2$ $\cdots$. Then, each instance of $\left(\frac{\varepsilon}{k-1}, \frac{\delta}{k-1}\right)$-PAC learning is performed. That is, PAC learning of binary models (i), (ii), (iii), $\cdots$ is simultaneously performed using $m_L^{(2)}\left(\frac{\varepsilon}{k-1}, \frac{\delta}{k-1}\right)$ training data. This enables the following relationships to be satisfied.

(i) The probability of $P_{\mathcal{D}}(f(u) = 0, h(u) \geq 1) + P_{\mathcal{D}}(f(u) \geq 1, h(u) = 0) \leq \frac{\varepsilon}{k-1}$ is $1 - \frac{\delta}{k-1}$ or more.
(ii) The probability of $P_{\mathcal{D}}(f(u) \leq 1, h(u) \geq 2) + P_{\mathcal{D}}(f(u) \geq 2, h(u) \leq 1) \leq \frac{\varepsilon}{k-1}$ is $1 - \frac{\delta}{k-1}$ or more.
(iii) $\cdots$

The probability that all PAC learning of these $k-1$ binary models will succeed, i.e., that the respective error rates will all be less than or equal to $\frac{\varepsilon}{k-1}$, is greater than or equal to $1 - \delta$. From this, it can be shown that

$$\begin{aligned}
&E_{\mathcal{D}}(|f(u) - h(u)|) \\
&= \sum_{i<j}(j-i)P_{\mathcal{D}}(f(u) = i, h(u) = j) + \sum_{i>j}(i-j)P_{\mathcal{D}}(f(u) = i, h(u) = j) \\
&= \sum_{i \leq l < j} P_{\mathcal{D}}(f(u) = i, h(u) = j) + \sum_{i > l \geq j} P_{\mathcal{D}}(f(u) = i, h(u) = j) \\
&\leq \sum_{l=0}^{k-2} \frac{\varepsilon}{k-1} = \varepsilon.
\end{aligned}$$

We obtain inequality (12) since $m_L^{(2)}\left(\frac{\varepsilon}{k-1}, \frac{\delta}{k-1}\right) = m_l^{(2)}\left(\frac{\varepsilon}{k-1}, \frac{\delta}{k-1}\right)$ when $k = 2$. □

Next, we consider PAC learning when the possible label values are extended to real values. Let $W = [a, b]$, i.e., the label $f(u)$ of each node is a real value in the closed interval of a and b, and let $\mathcal{H}$ be the set of functions from V to W that satisfy condition (10). In this setting, as shown below, PAC learning is possible even though $|\mathcal{H}| = \infty$. The minimum number of samples for which the probability of selecting $h \in \mathcal{H}$ satisfying $E_{\mathcal{D}}(|f(u) - h(u)|) \leq \varepsilon$ is more than $1-\delta$ is $m_l^{[a,b]}(\varepsilon, \delta)$. In this case, the following theorem can be obtained compared with that for the binary model.

Theorem 4.

$$m_l^{(2)}\left(\frac{\varepsilon}{b-a}, \frac{\delta}{b-a}\right) \leq m_l^{[a,b]}(\varepsilon, \delta) \leq m_l^{(2)}\left(\frac{\varepsilon^2}{2(b-a)}, \frac{\varepsilon\delta}{b-a}\right) \tag{13}$$

Proof. Divide the interval $W = [a, b]$ into $K = \lceil \frac{b-a}{\varepsilon} \rceil$ intervals of width ε or less: $[a, a_1], (a_1, a_2], (a_2, a_3], \ldots, (a_{K-1}, b]$. If $a_0 = a, a_K = b$, then, for $1 \leq i \leq K$, $a_i - a_{i-1} \leq \varepsilon$. We shall guess which interval the value of $f(u)$ at each node u is in and determine that, if $f(u) \in (a_{i-1}, a_i]$, then $h(u) = \frac{1}{2}(a_{i-1} + a_i)$. Similar to the proof of Theorem 3, we define the following $K-1$ binary models: $[a_0, a_1]$ or $(a_1, a_K]$, $[a_0, a_2]$ or $(a_2, a_K]$, $\cdots$, $[a_0, a_{K-1}]$ or $(a_{K-1}, a_K]$. For these $K-1$ binary models, $\left(\frac{\varepsilon}{2(K-1)}, \frac{\delta}{K-1}\right)$-PAC learnings are performed simultaneously. The probability that all the PAC learnings successfully estimate $f(u)$ within the range of error $\frac{\varepsilon}{2(K-1)}$ is $1-\delta$ or better. The error rate in this case is within $\frac{\varepsilon}{2}$. In addition, the difference between a point within each interval and the midpoint of that interval is within a maximum of $\frac{\varepsilon}{2}$, which means that the error rate is kept within ε, the sum of these values. □

6 Dimensions in Directed Graphs

The VC, Natarajan, and Graph dimensions are introduced as measures of the complexity of the set $\mathcal{H}$ of all possible labelings on V. The values of these dimensions are important when considering PAC learning, and in particular, whether these values are finite is a criterion for PAC learnability. According to Theorem 2, the Natarajan and Graph dimensions are used to estimate the upper and lower bounds on the amount of training data required for a multi-level model. As a comparison between these dimensions, inequality (9) holds as described in Section 3.4. Therefore, the evaluation of (8) can be expressed in terms of the Natarajan dimension only. It is known that the VC dimension of $\mathcal{H}$ satisfying condition (10) in the case of binary labels is determined from graph G as (11). Here, we show that the Natarajan and Graph dimensions of $\mathcal{H}$, which are defined as the set satisfying condition (10) in the case of multi-valued labels, can be specifically determined from G. This enables the upper and lower bounds on

the amount of training data required for PAC learning to be expressed in terms of the properties of graph G.

Proposition 1. *The following holds for $\mathcal{H}$, which is defined as the set satisfying condition (10) on directed graph $G = (V, E)$.*

$$d_N^{(k)}(\mathcal{H}) = d_G^{(k)}(\mathcal{H}) = \max_{U \subseteq V}\{\ |U| \mid {}^{\forall}u, v \in U, \rho_U(u, v) \neq k-1\} \tag{14}$$

Since the dimensions that can be determined as above are the values known from directed graph G, we will write them as $d_N^{(k)}(G)$ and $d_G^{(k)}(G)$. Note that condition $\rho_v(u, v) \neq k-1$ on the right-hand side of Equation (14) means that no directed edge sequence of length $k-1$ or more is contained within a subset U of V. Note that this coincides with (11) when $k = 2$. Also note that, in the k-valued model on directed graph G, the value of $d_{VC}^{(k)}(\mathcal{H})$ described in Section 3.2 coincides with $d_{VC}(G)$ defined by (11). Putting these together, we obtain

$$d_{VC}(G) = d_N^{(2)}(G) = d_G^{(2)}(G) \leq d_N^{(k)}(G) = d_G^{(k)}(G). \tag{15}$$

Proof. It suffices to show the following. (i) If U is a subset V and contains no directed edge sequence of length $k-1$ or more, then U is N-shattered by $\mathcal{H}$, which is determined from G. (ii) When U is a directed edge sequence of length $k-1$ or more in G, U is not G-shattered by $\mathcal{H}$ determined from G.

(i) Let U be a subset of V that contains no directed edge sequence of length $k-1$ or more. For each point u in this U, define $\beta(u)$ as

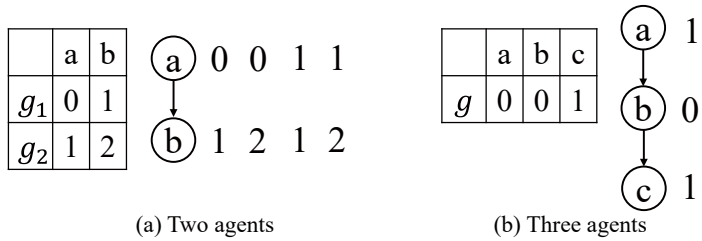
$$\beta(u) = \text{Maximum length of directed edge sequence ending at } u \text{ in } U,$$

where $0 \leq \beta(u) \leq k-2$. Using this β, we define $g_1(u) = \beta(u), g_2(u) = \beta(u) + 1$. Then $0 \leq g_1(u) < g_2(u) \leq k-1$, and $g_2(u) = g_1(u) + 1 \leq g_1(v)$ for any $u, v \in U$ satisfying $u \to v$. Let h be defined as follows. For any subset T of U, $h(u) = g_1(u)$ if $u \in T$ and $h(u) = g_2(u)$ if $u \notin T$. Such an h is contained in $\mathcal{H}$ determined from G because it satisfies $h(u) \leq h(v)$ if $u, v \in U$ and $u \to v$.

(ii) Suppose $U = \{u_0, u_1, u_2, \ldots, u_{k-1}\}$ and $(u_{i-1}, u_i) \in E$ for $1 \leq i \leq k-1$. Also suppose that function g from U to $\{0, 1, \ldots, k-1\}$ is contained in $\mathcal{H}$; that is, if $i \leq j$, then $g(u_i) \leq g(u_j)$ is satisfied. If we define a subset T of U from this g, as explained below, there does not exist an $h \in \mathcal{H}$ satisfying $h(u) = g(u)$ if $u \in T$ and $h(u) \neq g(u)$ if $u \notin T$.

(I) Let $u_0 \in T$ if $g(u_0) > 0$. Let $u_0 \notin T$ if $g(u_0) = 0$.
(II) For $1 \leq i \leq k-1$
(a) Let $u_i \in T$ if $u_{i-1} \in T$ and $g(u_{i-1}) < g(u_i)$.
(b) Let $u_i \notin T$ if $u_{i-1} \in T$ and $g(u_{i-1}) = g(u_i)$.
(c) Let $u_i \in T$ if $u_{i-1} \notin T$ and $g(u_{i-1}) + 1 < g(u_i)$.
(d) Let $u_i \notin T$ if $u_{i-1} \notin T$ and $g(u_{i-1}) + 1 = g(u_i)$.
(e) Let $u_i \in T$ if $u_{i-1} \notin T$ and $g(u_{i-1}) = g(u_i)$.

For T defined in this way, let $\mathcal{H}_0$ be the set of functions h from U to $\{0, 1, \ldots, k-1\}$ that satisfy the following conditions: $h(u_i) \leq h(u_j)$ if $i \leq j$,



	a	b
g_1	0	1
g_2	1	2

(a) Two agents

	a	b	c
g	0	0	1

(b) Three agents

Fig. 5. Example of calculating Natarajan dimension on basis of Proposition 1

$h(u) = g(u)$ if $u \in T$, and $h(u) \neq g(u)$ if $u \notin T$. We show that $\mathcal{H}_0$ is an empty set.

If $\mathcal{H}_0$ is not an empty set, we define $\tilde{h}(u_i) = \min_{h \in \mathcal{H}_0} h(u_i)$. If $i \leq j$, then $\tilde{h}(u_i) \leq \tilde{h}(u_j)$, so $\tilde{h}$ is also included in $\mathcal{H}_0$. From (I), $\tilde{h}(u_0) \geq 1$; furthermore, $\tilde{h}(u_0) = 1$ if $u_0 \notin T$. From the way (b) and (d) are determined, we know by induction with respect to i that $\tilde{h}(u_i) = g(u_i) + 1$ if $u_i \notin T$. Therefore, if $u_i \in T$ is determined by (e), then $\tilde{h}(u_{i-1}) > g(u_{i-1}) = g(u_i) = \tilde{h}(u_i)$, which is a contradiction. When $u_i \in T$ or $u_i \notin T$ is determined by (a), (b), (c), and (d), $\tilde{h}(u_{i-1}) < \tilde{h}(u_i)$ for $1 \leq i \leq k-1$. Therefore, $\tilde{h}(u_{k-1}) > \tilde{h}(u_0) + (k-1) > k-1$, which violates the assumption. □

We show how we calculate the Natarajan dimension in the following example.

Example 6. Let X be $\{a, b, c\}$ and W be $\{0, 1, 2\}$. The structure of the social network is a line in which the edge from agent a goes to agent b and the edge from agent b goes to agent c. Here, since $k = 3$, we determine the maximum number of agents who do not have an edge sequence of length $2(= k-1)$ or more.

For agents a and b, when we determine g_1 and g_2 as shown in (a) in Figure 5, any assignment of g_1 and g_2 to agents a and b is monotonically non-decreasing on the path. Thus, $S = \{a, b\}$ is N-shattered by hypothesis $\mathcal{H}$. For agents $S = \{a, b, c\}$, we define g as shown in (b) in Figure 5 for example. If we set $T = \{b, c\}$, then there is no h that satisfies $h(x) = g(x)$ for $x \in T$, $h(x) \neq g(x)$ for $x \in S \backslash T$ and is monotonically non-decreasing on the path. Thus, the Natarajan dimension is 2.

As a result, Theorem 2 can be rewritten in terms of the structure of directed graph G as follows.

Theorem 5. *When function f from V to $W = \{0, 1, \ldots, k-1\}$ on directed graph $G = (V, E)$ satisfies condition (10), we assume that the value $d_N^{(k)}(G)$ defined by Equation (14) is finite in PAC learning to guess this f. In this case, there exist positive constants C_1 and C_2 independent of $d_N^{(k)}(G), \varepsilon, \delta$, and the*

following holds.

$$C_1 \frac{d_N^{(k)}(G) + \log(1/\delta)}{\varepsilon} \leq m_L(\varepsilon, \delta) \leq C_2 \frac{d_N^{(k)}(G) \log(1/\varepsilon) + \log(1/\delta)}{\varepsilon} \tag{16}$$

Moreover, in this model of a directed graph, we see that $d_N^{(k)}(G)$ has an upper bound depending on the value of the VC dimension determined from G, as in the proposition below.

Proposition 2. *For a hypothesis class $\mathcal{H}$ defined as a set satisfying condition (10) on directed graph $G = (V, E)$, the following holds.*

$$d_N^{(k)}(G) \leq (k-1) d_{VC}(G) \tag{17}$$

From this proposition, it follows that models on directed graphs cannot have $d_N^{(k)}(G) < \infty$ and $d_{VC}(G) = \infty$. Therefore, it can be seen that a finite width of the partially ordered set corresponding to a directed acyclic graph is a necessary and sufficient condition for PAC learnability.

Proof. Let U be a subset of V that contains no directed edge sequence of length $k-1$ or more. If we define $\beta(u)$ for each node u in U as in the proof of Proposition 1, then $0 \leq \beta(u) \leq k-2$. From this definition, if $\beta(u) = \beta(v)$ for $u, v \in U$, then $u \not\to v$ and $v \not\to u$ in U. Let $U_l = \{u \in U | \beta(u) = l\}$ for $0 \leq l \leq k-2$. Each U_l is an antichain, and at least one U_l satisfies $|U_l| \geq \frac{1}{k-1}|U|$. Therefore, $d_{VC}(G) \geq \max_l |U_l| \geq \frac{1}{k-1}|U|$ holds. □

7 Conclusion

We used the PAC learning framework to determine the number of samples needed to estimate the degree of propagation of multiple opinions in a social network. First, we compared the binary and multi-valued opinion cases. Next, we showed that the Natarajan dimension in this problem setup is determined by the size of the largest subset that does not contain directed edge sequences of length $k-1$ or more. We showed that the required number of samples can be estimated from above and below by using the Natarajan dimension. Furthermore, when the number of nodes in the graph is infinite, PAC learnability is determined by the structure of the graph, and the condition is the same for the binary and multi-valued cases.

Future work includes estimating the required sample size when the social network structure known to the learner is incorrect, when there is an agent that propagates a false opinion, and when agents follow the majority opinion upon receiving multiple opinions.

Acknowledgments

This work was partially supported by JSPS KAKENHI Grant Numbers JP18H03299, JP18H03337, JP21K12191, and JP21K19833 and by JST CREST Grant Number JPMJCR21D1.

References

1. Balcan, M-F., Procaccia, A.D., Zick, Y.: Learning cooperative games. In: Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI-2015). pp. 475–481 (2015)
2. Ben-David, S., Cesa-Bianchi, N., Haussler, D., Long, P.M.: Characterizations of learnability for classes of $\{0, \cdots, n\}$-valued functions. Journal of Computer and System Sciences **50**, 74–86 (1995)
3. Berenbrink, P., Hoefer, M., Kaaser, D., Lenzner, P., Rau, M., Schmand, D.: Asynchronous opinion dynamics in social networks. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2022). pp. 109–117 (2022)
4. Conitzer, V., Panigrahi, D., Zhang, H.: Learning opinions in social networks. In: Proceedings of the 37th International Conference on Machine Learning (ICML-2020). vol. 119, pp. 2122–2132 (2020)
5. Daniely, A., Sabato, S., Ben-David, S., Shalev-Shwartz, S.: Multiclass learnability and the ERM principle. Journal of Machine Learning Research **16**, 2377–2404 (2015)
6. Felsner, S., Raghavan, V., Spinrad, J.: Recognition algorithms for orders of small width and graphs of small Dilworth number. Order **20**, 351–364 (2003)
7. Igarashi, A., Sliwinski, J., Zick, Y.: Forming probably stable communities with limited interactions. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI-2019). pp. 2053–2060 (2019)
8. Irfan, M.T., Hancock, K., Friel, L.M.: Cascades and overexposure in social networks: The budgeted case. In: Proceedings of the 21st International Conference on Autonomous Agents and Multiagent Systems (AAMAS-2022). pp. 642–650 (2022)
9. Procaccia, A.D., Rosenschein, J.S.: Learning to identify winning coalitions in the PAC model. In: Proceedings of the 5th International Joint Conference on Autonomous Agents and Multiagent Systems (AAMAS-2006). pp. 673–675 (2006)
10. Shalev-Shwartz, S., Ben-David, S.: Understanding Machine Learning - From Theory to Algorithms. Cambridge University Press (2014)
11. Sliwinski, J., Zick, Y.: Learning hedonic games. In: Proceedings of the 26th International Joint Conference on Artificial Intelligence (IJCAI-2017). pp. 2730–2736 (2017)
12. Zhang, H., Conitzer, V.: Incentive-aware PAC learning. In: Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI-2021). pp. 5797–5804 (2021)