# The Hat Problem: A New Approach to Transductive Learning?

Siyu Zeng

September 13, 2025

**Abstract**

This paper investigates the connections between transductive learning and the hat problem in mathematics. Transductive learning is currently a relatively unexplored area of research used in statistical learning that can directly make predictions on a set of test data without learning a function and generalizing it. The other subject of this paper, the hat problem, is a classic combinatorial puzzle that bears interesting similarities to the transductive model. There are many versions of this problem, which are also explored in the paper, along with their connections with the transductive learning model. Using different variations of the hat problem, the paper will show how different aspects of the hat problem are very much analogous to the transductive learning model, including connections between their setups, graphical representations, and solutions. The paper also discusses some limitations of this analogy from key differences between transductive learning and the hat problem. Additionally, based on these connections, it proposes future steps in this novel field of research.

**Keywords:** Adversary, Bipartite matching, Codeword, Hamming space, Prediction strategy, One-inclusion graph

## 1 Introduction

While most research in the machine learning field has been focused on inductive learning, transductive learning has become just as significant in importance. The transductive learning model was originally proposed by Vapnik, emphasizing a "particular to particular" idea [11]. The transductive model is one that, unlike inductive models, does not need to infer a function from the training data before making predictions on the test data, thus skipping the intermediate step of generalizing. Instead, it uses strategies that can be either deterministic or random to predict the missing labels of the testing data points directly. As it has gained popularity in recent years, more perspectives on it have been unravelled. In previous research, transductive learning models have been compared to the Probably Approximately Correct (PAC) learning model in the work of Dughmi et al.[6] and Haussler et al. [9], which provided insights allowing us to utilize simpler techniques. However, no work has been done to make an analogous comparison with the hat problem.

In this research paper, a novel approach to the transductive learning model will be taken by comparing it to a classical puzzle in recreational mathematics, the hat problem. Specifically, it will focus on discussing a special case of transductive learning that has been used in prior work, which involves a single unlabeled data point for prediction. This model exists as a "game" between two players: an *adversary* chooses a set of labelled data points and covers a single data point for the learner to guess, and the learner has to find the best strategy to predict the data point. This is essentially analogous to the hat problem, which is also a similar 2-player game. This paper will proffer to the reader the idea that for both the setup, graphical representations, and certain aspects of the solution, the hat problem could be considered "close enough" to the transductive model. Because of the considerable number of correlations that exist between them, an examination through the lens of the hat puzzle could yield alternative insights in learning theory, particularly for classification problems. In turn, these new insights could be extremely beneficial for future development and efforts in improving the transductive model.

### 1.1 Outline

To propound the idea that the transductive model is the hat problem to an extent, this paper will first provide the basic background of the transductive model and the hat problem. Then, it will connect the analogous elements of the hat problem and the transductive model, discussing their respective setups and graphical representations. Subsequently, utilizing different variations of the hat problem, further connections to the details regarding the transductive model will be established. Then, it will highlight some key differences creating limitations on this analogy. Lastly, based on these connections, this paper will suggest how strategies from the hat problem could be applied to approach transductive learning in an alternative way, to speculate future research steps.

## 2 Background

In this section, the paper will recap the basics of both the hat problem and the specific transductive model that the subject will be dealing with. Solutions and variations of the hat problem will also be outlined.

### 2.1 The hat problem

The original version of the hat problem, called the "seven prisoners puzzle," was proposed by Ebert, and more about it can be found in his work on complexity theory [15]. Here, the paper will start by detailing a fewer-player version of this game and subsequently extend it to more generalized variations.

Consider three prisoners sitting in a room who are playing a guessing game. If they win, they get the chance of collective release. The game starts with someone placing either a red hat or a blue hat on each of the prisoners' heads (the hat could be randomly placed or placed by an adversary). Each prisoner cannot see the hat on their own head but can see the hat colors of the two other prisoners. Then, they are all prompted to simultaneously make a silent guess regarding their own hat color, in which case, each of them can either make a guess of red or blue, or pass (not guess). Throughout the process, no communication of any kind is allowed between them, which means that the prisoners can only base their guesses on the hat colors they see on the other prisoners. Ultimately, the prisoners will win collectively if at least one prisoner guesses his own hat color correctly, with no prisoner guessing incorrectly (players can skip their turns). The question is, how can they find a strategy to maximize their chances of winning [9]?

The solution to this problem is surprisingly clever. Before entering the game, the prisoners can coordinate a guessing strategy to help them win in the majority of possible scenarios. The strategy they use is that if a prisoner sees two other hats of the same color, he would guess that his own hat color is the opposite. Otherwise, he would pass on his turn[7].

The chances of the prisoners winning can be visualized by looking at all the possible configurations illustrated in the table below. There are eight possible hat placements in total (Table 1). The prisoners only lose when all three of them have the same colored hats, in which case, all of them will guess the incorrect color. Since they only lose in two of the eight possible cases, they win in $\frac{3}{4}$ of the total scenarios, which gives them a 75% chance of winning [7]. Notice that all six possible incorrect guesses from the players are concentrated into the two losing scenarios to give the fewest number of losing cases as possible.

| P1 | P2 | P3 | Result |
|---|---|---|---|
| R | R | R | Lose |
| R | B | R | Win |
| R | B | B | Win |
| R | R | B | Win |
| B | B | B | Lose |
| B | R | B | Win |
| B | B | R | Win |
| B | R | R | Win |

Table 1: 8 possible configurations of hat colors and their results

The setup of this game could also be converted into a sequence of binary digits. The colors red and blue can be represented as the digits 0 and 1, respectively, and each of the hats on the players' heads will be represented with a binary digit. The room of three prisoners would thus be a sequence of three digits $d$, each representing the colors of the three hats. For example, 000 corresponds to all three prisoners wearing red hats. The index $i$ of each digit corresponds to a unique player, and what a player sees would be a sequence of all other digits $\{d_j : j \neq i\}$, with their own hat color masked as a question mark.

The strategy then becomes a numbers game. If a player sees two of the same digits, he would guess his own digit to make the resulting sequence of digits non-uniform. That is, if they see two other 0s, he would guess his digit to be 1, making the entire sequence of two 0s and one 1, and vice versa [7].

## 2.2 Generalization to the N-Hats Problem

The three-person case of the hat problem can be generalized to $n$ players, where $n$ is an integer greater than 3. For more than three players, this problem can be rewritten more abstractly. This abstraction will be shown in the following passages.

As discussed previously, the hat game consists of a sequence of digits representing the hat colors of each player: $(y_1, y_2, y_3, ..., y_n)$, where each $y_i$ is either 0 or 1. This sequence can alternatively be viewed as n-dimensional vectors over the field $\mathbb{F}_2 = \{0, 1\}$[1]. Re-framing the game setup this way allows one to remove any non-essential elements from the central idea, and to think of it mathematically.

Now, it is possible to come up with a generalized solution for this generalized version of the problem. In the hats game, there are in total $2^n$ permutations of sequences of 0s and 1s. These $2^n$ sequences can each be thought of as a vector with $n$ entries, and they together form a vector space $\mathbb{F}_2^n$, called a *Hamming space*, which is shown in Figure 1. Essentially, a Hamming space contains the set of possible hat color configurations for an $n$-player game [8].

For $n$ people, this game is modelled as a vector consisting of $n$ entries that each represent a hat color for a different player. From the perspective of a single player in the $i$th position, he would know the labels of $n-1$ entries in the vector with certainty but have an ambiguity regarding the entry $y_i$, which is his own label. Specifically, the vector representing a player's perspective could look like $(y_1, y_2, y_3, ..., ?, ...y_{n-1}, y_n)$, with the question mark representing the ambiguous label. When a player is unsure of this label, he does not know whether the correct configuration of hats is $(y_1, y_2, ...0, ..., x_n)$ or $(y_1, y_2, ...1, ..., y_n)$ [4].

Therefore, the ambiguity of labels is analogous to being situated on an edge of the Hamming space, where it connects the two possible vertices associated with the missing label. Making a prediction would correspond to pointing this edge toward one of the two vertices, and pointing the edge toward the correct vertex would correspond to making the correct prediction [13]. Effectively, this turns the game into a matter of directing the edges in the Hamming space to give the optimal solution. Different variations of the game will correspond to different strategies, but they can all be reduced down to some way to orient edges on the cube.

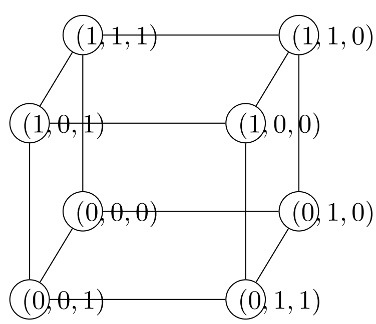


Figure 1: A three-dimensional Hamming space

Now it is possible to generalize the strategy from the three-person game. Strategies in the game deal with subspaces of the Hamming space. Choosing a certain subspace $C$, this subspace will be defined to be all the *codewords* of this game. The codewords are a set of vectors pre-decided by the players before they enter the room, and which would determine what a player guesses during the game. Having codewords allows us to define certain "losing" scenarios for the players. This is because, in the strategy, players in every possible scenario will predict that they are not in the cases specified by the codewords[8]. The strategy is optimal when all non-codeword nodes are connected to a codeword, which directs all of them away from the losing scenario. However, this only occurs when the number of players in the game is $2^m - 1$, where $m$ is an integer [3], because that is when every non-codeword can be adjacent to a codeword. Otherwise, the strategy cannot achieve the minimum possible number of mistakes for $n$ players, which is $\frac{1}{2^m}$.

Notice that the codewords in the three-person game have a Hamming distance of 3 from each other. This allows every other vertex on the cube to be connected to the codewords by an edge. This gives us a way to construct the guessing strategy for the three-person game. First, the subspace $C$ will be defined to contain the codewords (000) and (111), which are the two cases in which the players lose. Then, the three edges incident to each codeword will be pointed away from it to create the final Hamming graph [4]. This orientation means that whenever the true configuration lands on a codeword, the learner will make a mistake.

When following this strategy, the $n$ players are instructed on how to guess their hat colors when they are on an edge contingent on a codeword. If this situation happens, they should always guess a digit to make the resulting vector not a codeword [7]. For example, in the 3-player case, if a prisoner sees a vector $(?, 0, 0)$, which is when a player sees two red hats, he would guess that his own hat label is 1 (a blue hat), which would make the vector $(1, 0, 0)$.

## 2.3 Hat Problem variations

There are many variations of the hat game. However, the central game strategies all center around the basic idea of a vector space of all the possible configurations.

### 2.3.1 Majority-hats game

The majority hats game has similar conditions for the players as the basic version of the hats game (i.e. players guess simultaneously, no communication is allowed). However, unlike Ebert's version of the game, the majority-hats game requires players to guess, and the goal is to maximize the probability that the majority of players guess their own hat color correctly. Here, the strategy is to form a closed cycle of three players $A, B$, and $C$, with each player guessing the opposite color of the player before him. This also yields a 75% win rate. The generalized strategy is to "guess the opposite as your neighbour"[4], as explained by Berlekamp.

In this version of the game, the deterministic strategy from each player's "perspective" can still be interpreted in the Hamming space as a problem of orienting edges. Nevertheless, the orientation of the edges would be slightly different. Previously, the edges other than those incident to the codewords would be undirected, indicating a pass on those players' turn. For the three-person case in the majority-hats game, the strategy would be modified for these undirected edges to form an Eulerian circuit among these same, non-codeword vertices [5].

### 2.3.2 Maximum correct hats game

The maximum correct hats game is similar to the majority hat game: in each instance, multiple people guess to try and maximize the number of correct answers. However, the difference in these two games lies in the objective of the players. In the majority hat game, the objective of the players is to maximize the number of cases where the players satisfy the majority-correct condition. On the other hand, the maximum correct hats game focuses on getting as many correct guesses as possible in one case. Furthermore, since an adversary is placing the hats, a *prediction strategy* must guarantee a certain number of correct guesses in every possible case. Note that the maximum correct hats game is also known as the adversarial hat game described by Butler et al. [5], which will be discussed further in section 5.

### 2.3.3 Limited hats game

The limited hats game is a variation where the adversary has restrictions on the number of hats of each color [6]. That is, instead of being able to pick freely from an infinite supply of all hat colors, it would "run out" of a hat color after a certain number of draws. For the mathematical representation, the set that contains the different hat colors could be expressed as $C = \{c_1, c_2, ...c_m\}$, where $c_i$ is the number of hats of the $i$th color. If $c_i = n$ for all $i$, this would be the same problem as if there were no restrictions. However, if there is any $c_i < n$, it is possible to narrow down its pool of possible hat configurations to allow for a more sophisticated strategy.

### 2.4 The Transductive model

This section details the background of the specific transductive model that will be used in this paper. This specific case involves only a single missing label in a dataset, which needs to be guessed via a single prediction.

One can start by setting up the model, which has a domain $X$ and labels from $Y$. The model tries to predict the label of a point $x \in X$ described by an unknown function $h$ based on known values of other points from $X$. In the transductive model, there are both deterministic and random strategies used to predict the label, but this paper will focus on deterministic strategies. One way to frame the transductive model is to think of it as a game between an adversary and a learner [9]. The game fulfills the following steps:

First, the adversary chooses the target dataset for the learner to guess, which is a set of points $x_1, x_2, ...x_n$ from the domain $X$ and a function $h$, which is in turn used to create the set of data points $(x, h(x))$. This set of data points represents the target predictions for the learner, the ground truth dataset. Then, the adversary creates another dataset by covering a label, $y_i$, for the learner to guess. Here, the learner faces a masked dataset $D = \{(x_1, y_1), (x_2, y_2), \ldots, (x_i, ?), \ldots, (x_n, y_n)\}$, with the question mark representing the label that is unknown to it. Subsequently, the learner needs to guess the unknown label, which results in a loss of $l(h(x), y)$. Note that while the adversary can control what set of points is contained in the entire dataset, it cannot choose which label is covered. That is, the position of the question mark will be chosen randomly. This paper will discuss when the learner uses a certain deterministic prediction strategy to guess the missing label. In other words, when provided with some hypothesis $h \in H$ that describes the data, the learner needs to look at the known labels of the other $n-1$ points in the dataset to find the best guess for the missing label based on the rest. The goal is to minimize the expected loss of the learner for the hypothesis class $H$ [9]. In other words, the learner tries to minimize

$$\Pr[y_i \neq h(x_i)].$$

#### 2.4.1 Two settings of the transductive model

There are two settings of the transductive model, the agnostic and the realizable setting [6]. In the realizable setting, the adversary has certain restrictions on what datasets it can choose. These restrictions are imposed by a hypothesis class $H$, and the learner guesses based on the assumption that some hypothesis $h \in H$ can perfectly classify the data with no errors. In other words, it assumes that for every $x_i$ from the dataset, $h(x_i) = y_i$. This setting allows the learner to make predictions with a relatively small subset of dataset configurations among the total $2^n$ possibilities. For example, if a dataset $(0, 0, ?, 0, 1)$ was perfectly separable by a halfspace, there would only be one possible scenario for the missing label because the possibility of $(0, 0, 1, 0, 1)$ is restricted by the hypothesis space. Thus the learner would have no hesitation to predict the question mark as 0. From this, it is easy to see that the loss would be calculated by looking at where the guess is not the same as the ground truth[6].

The other setting of this model is the agnostic setting. Unlike the realizable setting of the transductive model, this setting would impose no restrictions on the adversary, which means that it is free to choose any arbitrary dataset. For the learner, it is thus necessary to consider any ground truth among the $2^n$ configurations as a possibility. Because there may not be a "perfect" hypothesis in this setting of the model, the learner tries to explain the dataset close to the best hypothesis. Therefore, the loss needs to take into account the deviation from the ground truth and is described as a loss relative to the best-in-class loss. Specifically, its error differs from the one in the realizable learning from subtracting the best-in-class loss $min_{h \in H} \sum l(h(x_i), y_i)$ [6].

Despite their differences, the goal of the transductive model in both the agnostic and the realizable settings is to minimize the transductive error rate and the sample complexity.

# 3 Connections

In the previous sections, the hat problem and the transductive model have been reduced down to their mathematical backbones. The following section will draw some connections between different variations of the hat problem and the transductive model. Besides explaining the similarities between the situations of both topics, it will also make analogies based on their respective graph representations.

## 3.1 Turning transductive learning into a hat problem

There are many connections between the setups of the transductive learning model and the hat problem. In the hat problem, an adversary places hats of certain colors on the players. In the transductive model, this can be thought of as the adversary choosing a ground truth dataset that reflects the target of the model's predictions. This chosen dataset can be represented as a sequence of elements $D = \{d_1, d_2, ..., d_n\}$, where $d_i$ is the data point at index $i$ (i.e. a hat at position $i$). Here, $d_i$ could be an element of any dimension. For example, it could be a scalar $d_i = y_i$ or a vector $d_i = (x_i, y_i)$. $D$ would represent the sequence of hat colors in the hat problem, but in the transductive model, each element in the sequence instead represents a label for a data point. For now, this paper will stick to discussing the simplest case of binary classification. This is when the adversary chooses every element $d$ from a 2-element set $C = \{0, 1\}$, and creates the dataset $D$ as a vector space over $\{0, 1\}^n$ [7].

Having chosen the ground truth dataset, the adversary then creates another set $D_{-i}$ where one label, $y_i$, is masked. This dataset is then shown to the learner, which is equivalent to how a set of hats on the other players are shown to one player in the hat game. Subsequently, the learner is asked for a prediction on the label based on the known data points in $D$. Similar to the hat problem, where the players each see a set of hat colors with one masked hat, the learner in the transductive model sees a dataset with a missing label. $D_{-i}$ precisely represents the "view" of the learner when looking at the other data points. When the learner is prompted to predict the missing label, it is analogous to a player in the hat problem trying to guess his own hat color as either 0 or 1 in a dataset with one unknown hat color, $(y_1, y_2, y_3, ..., ?, ...y_{n-1}, y_n)$. Simply, the learner's game with the adversary could be translated into predicting a {0,1}-valued function on $\mathbb{F}_2^n$, where $n$ is the number of players in the game or the number of data points in the dataset[7]. In summary, both problems involve a prediction of a masked label from known data points based on a prediction strategy. From this interpretation, a learner in the transductive model predicting the label $i$ from a set of known data points $D_{-i}$ is equivalent to a player in the hat problem guessing his own hat color.

Both problems aim to come up with a prediction strategy to maximize the win rate. A prediction strategy in the hat game is, by formal definition, a collection of strategies $S$ with cardinality $n$, where each strategy corresponds to one of the $n$ players [17]. Each singular strategy $s_i$ for player $p_i$ is a function $s_i : C^{n-1} \to C$, which takes the input set of categories and outputs a guess about $p_i$'s hat color [17]. Then, the learner generates a prediction regarding the label of that data point. Neither the hat problem nor the transductive model needs to infer a function to describe the training set first, but can instead directly label the test set with prediction strategies. From this interpretation, the strategies in the hat game and the transductive model can be formalized similarly. This paper focuses on where each $s_i$ is a deterministic function, meaning that it has a definite output based on a function of the $n - 1$ known data points as the input.

For some versions of the hat game, the hats are chosen by an adversary, which is someone who knows the prediction strategy of the players and will place the hats accordingly to induce the most failure for them. Similarly, the adversary in the transductive model can be analogized to someone who always tries to increase the error of the label's prediction by choosing a dataset that would create the most error when the learner predicts. This situation makes it so that when trying to decide on a strategy, the players need to create one that minimizes the error across all configurations of the hat colors. Therefore, it is also important in the transductive model to consider the error in all possible configurations for a vector space over $\mathbb{F}_2^n$. More about this adversarial variation of the hat problem will be discussed in section 5.

### 3.1.1 Graphical representations

Being of very similar natures, the transductive learning model can be visualized in the same graphical fashion as the hat problem. In the hat problem, these vectors are equivalent to the $2^n$ possible configurations of hat colors, which can be represented as a coordinate in the Hamming space to form an $n$-dimensional cube. In the transductive model, all the possible configurations of the dataset, the ground truths, can be represented as a hypercube in the *one-inclusion graph*. Similar to the problem, each vector representing a ground truth dataset occupies that coordinate in the $n$-dimensional space [4]. For example, consider the dataset $D$ with length $n$ and labels chosen from $\{0, 1\}$. This dataset has $2^n$ possible configurations of sequences, each representing a possible ground truth of the data. The $2^n$ possible ground truths can each be visualized as a coordinate in the $n$-dimensional space. One of these nodes could be the vector $(0, 0, 1)$, which occupies this exact coordinate in the OIG (one-inclusion graph). Every two nodes in the set of possible ground truths that differ by only one coordinate are connected with an edge. Together, they form an $n$-dimensional hypercube with the ground truth as vertices and edges connecting them.

Representing the ground truths this way is very useful, as it can lead to many interesting interpretations of the details in transductive learning. For instance, in the hat problem, if a player does not see his own hat color, this corresponds to an ambiguity about that particular label in the dataset. This ambiguity is represented by an edge on the Hamming graph. Similarly, the edges of the hypercube can be seen as an uncertainty that the learner has, which occurs when the masked dataset with $n-1$ elements is shown to the learner. Each edge represents a vector with a masked label at index $i$, or the dataset $D_{-i}$. Like in the hat problem, guessing a label in the transductive model will orient the corresponding edge in a direction, turning the OIG into a directed graph [4]. To further illustrate this, observe that when one label is covered, there are two possible ground truth datasets for the learner to distinguish: $D_{0,i}$ and $D_{1,i}$, where $D_{\chi,i}$ is a dataset with a predicted label $\chi$ at index $i$. Thus, the learner is positioned on an edge between the two possible ground truth vertices. Each guess made by the learner corresponds to orienting the edge towards the direction with the vertex it decides. If the prediction is correct, the edge will be oriented towards a vertex representing the ground truth. On the other hand, the learner makes an error when the edge is pointed away from the ground truth vertex. Thus, by comparing the transductive model to the hat problem, it can be translated into a problem of directing the edges on the OIG.

## 3.2 The Realizable Setting

The two settings of the transductive model will have slightly different graphical representations. Due to differences in their constraints, the one-inclusion graph will be different in the realizable and agnostic settings. Recall that in the realizable setting, the adversary is restricted so that the ground truths can be perfectly explained by some hypothesis $h \in H$. Because of this, the one-inclusion graph can be modified so that certain vertices that cannot be perfectly classified (i.e. the "impossible" ground truths) are removed from the hypercube [6]. For example, a possible OIG for a realizable transductive model with three data points restricted by a certain hypothesis class $H$ could look like Figure 2. This scenario could be compared to the limited hats variation of the hat problem: because there are limitations on how many hats of each color the adversary can choose, there are certain "impossible" nodes, which are removed from the Hamming graph. Similar to the limited hats game, the realizable setting would similarly constrain the prediction possibilities to a smaller subset of ground truths, and thus induce a better strategy for guessing. In fact, the number of nodes in a one-inclusion graph for the realizable setting can only be as large as $n^{O(VCdim(d))}$, where $d$ is the data set used [16]. More on the limited hats game will be discussed in section 5.

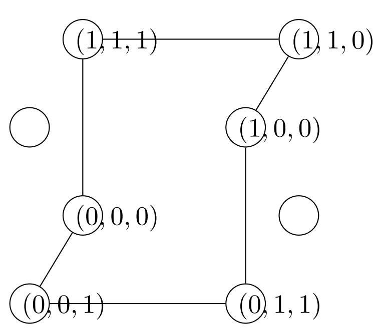


Figure 2: One-inclusion graph restricted by hypothesis class $H$

In the realizable setting, there are two possible scenarios during a learner's prediction: the edge representing the learner's perspective could either be incident to only one node or more than one. In the former, the incident node would be the only possible ground truth for the dataset $D_{-i}$. Thus, if the dataset given to the learner corresponds to that node, the prediction could immediately be made with no hesitation. In the latter scenario, there is more than one possible ground truth based on the masked dataset that the learner sees. In that case, the possible datasets would be represented as adjacent vertices in the one-inclusion graph, differing only by the entry at position $i$. From here, it is easy to see that the two vertices are connected by an edge representing the learner's ambiguity, and a solution again corresponds to an orientation of the edge.

### 3.2.1 A note on codewords

In the hat problem, a codeword is used as a "standard" on which optimal strategies are based. For example, the optimal strategy in Ebert's hat game is formulated based on orienting all edges adjacent to the codewords outwards [15]. Thus, in this hat problem, the codeword can be used as an analogy to the ground truths to assess the accuracy of the transductive model. To see this, note that if an edge is directed out of a vertex containing the ground truth, it is an error made by the learner. To calculate the error rate in the transductive model, the out-degree of the worst-case ground truth must be divided by $n$. Thus, the out-degree of the ground truth is proportional to the error rate of the transductive learner. For the hat problem, if an edge is directed toward a codeword, it corresponds to a "suboptimal" strategy, which causes the win rate to no longer be the largest possible. It is thus valid to say that in the hat problem, the accuracy of the players is proportional to the out-degree of the codewords. Essentially, the in-degrees and out-degrees of the ground truth serve as an important indicator of the optimality of the predictions [6]. More will be elaborated in sections 4 and 5.

## 3.3 The Agnostic Setting

In the agnostic setting, on the other hand, the adversary is not restricted to choosing a dataset perfectly describable by a hypothesis $h \in H$. Thus, there are no hard limitations on the set of possible ground truths. Instead, the adversary could choose any arbitrary dataset. This means that the possibilities of all $2^n$ ground truths configurations are preserved. In this sense, the agnostic setting of the transductive model is more similar to most hat problems discussed previously in this paper. In the variations of the hat problem seen above, there are similarly few restrictions on the possible configurations of hat colors. Consequently, for the OIG in this setting, it is not necessary to remove "impossible" nodes.

In the agnostic setting, since all ground truths are possible, the hypothesis is only used as a benchmark for calculating the loss[6]. From a graphical perspective, the loss considers the Hamming distance between the learner's prediction and the best possible hypothesis. When calculating the loss, the one-inclusion graph is a useful representation because it provides a visual representation of the distance from the best answer. To calculate how much a learner deviates from the best possible prediction, there is a loss of

$$\mathbb{E}\left[\ell(h(x_i), y_i)\right] - \min_{h \in H} \sum \ell(h(x_i), y_i).$$

Besides calculating the error, many hat problem variations aim to minimize the deviation from the correct configuration, which demonstrates further the utility of this graph.

## 3.4 Multiclass classification

The classification tasks of transductive learning models are not limited to binary categories. Multiclass classification also serves as an important area in many learning models, and it has its analogies with the hat problem as well.

Multiclass classification can be compared analogously to one particular variation of the hat problem: one that involves multiple colors of hats. The formalization of the multiple-colored hat game only requires one change to the setup in the number of colors. In this variation of the problem, instead of representing the colors of hats with colors chosen from $\{0, 1\}$, the set of $m$ hat colors in the game can be defined as $C = \{c_1, c_2, ..., c_m\}$, where $c_i$ is the $i$th hat color. The total number of configurations of hats is thus $m^n$, where $n$ is the number of players.

Similarly, instead of having only two colors, multiclass classification is equivalent to involving multiple colors in the hat problem. In this case, the adversary is able to choose its data points from the $m$ categories $C = \{c_1, c_2, ...c_m\}$. It thus chooses a ground truth dataset $D$ over $\{c_1, c_2, ...c_m\}^n$.

In the agnostic setting of a multiclass classification model, the one-inclusion graph would now have $m^n$ possible configurations. According to Dughmi, to graph the possible ground truths as nodes in the one-inclusion graph, the edges can be modified to be hyper-edges that pass through multiple nodes [3]. Like before, each hyper-edge would connect the nodes that share all the consistent $n-1$ data points, differing only on the point that is unknown to the learner. Because of this, for every $c \in C$, the nodes $(y_1, y_2, ..., c, , ...y_n)$ would exist on the same edge. As before, this can be thought of as a problem of orienting edges on the OIG, except this time the orienting is for hyper-edges.

# 4 Applying the Hat Problem to Transductive Learning

Now that the mathematical setup and abstractions of the transductive learning model have been effectively transformed into the hat problem, this paper will go into more detail about the similarities between them. In particular, this section will explore the applications of these similarities to the strategies, errors, and differences.

## 4.1 Bipartite Matching

Applying the analogies from the hat problem, one can make connections between the decisions in the OIG and a bipartite graph, which is an alternative representation of the OIG described in the work of Asilis et al. [3]. Subsequently, the problem of directing the edges on the OIG becomes a *bipartite matching* problem. Aggarwal et al. [1] described an intriguing algorithm to turn the hat problem into a bipartite matching problem, which was used to design deterministic algorithms for auction predictions. In this section, the hat problem will be used as an analogy to connect to the bipartite matching strategy for the transductive model.

Constructing this method starts with creating two sets of nodes. One set, $V$, contains the possible vectors that represent the ground truth dataset. This set can be placed on the right side of the bipartite graph. Let the set on the left, $V'$, contain all possible vectors $v_{-i}$, where $i$ is the position with the unknown hat color. These two disjoint sets make up the bipartite graph that the problem starts with. Each node $v_{-i} \in V'$ has arcs mapping to the $m$ possible nodes in $V$ that correspond to the possible hat configurations associated with $v_{-i}$, where $m$ is the number of hat colors in the game. As an example, the bipartite graph for a hat problem from the work of Aggarwal et al. can be shown in Figure 3. [1]. The figure shows a set $V'$ containing the vectors corresponding to the views of each of the players, and a set $V$ containing the vectors corresponding to the possible hat color configurations. Each node $v_{-i}$ has $m$ arcs mapping to nodes $v_{\chi,i}$, where $\chi$ is the predicted color at position $i$. These $m$ arcs represent all the possible guesses that a player faced with $v_{-i}$ can make.

In the transductive learning problem, using this analogy, the bipartite graph can be defined with two sets of nodes, $V'$ and $V$, where $V'$ contains the datasets $D_{-i}$ shown to the learner, and $V$ contains all the possible configurations $D_{\chi,i}$. The nodes in the set $V'$ represent the perspectives of each learner, and the nodes in $V$ represent the possible ground truth datasets. Each node from $V$ would have $m$ outgoing arcs to a node in $V'$, where $m$ is the number of classes for the classification problem. Following an arc from $D_{-i}$ to $D_{\chi,i}$ corresponds to predicting the label at $i$ to be $\chi$. Thus, to find a prediction strategy, it merely becomes a problem of finding a valid set of arcs from $V'$ to $V$ in the bipartite graph. From the $m$ outgoing arcs from each node on the left, a solution would require finding a subset of edges in the graph that has no more than one incident edge for each node. That is, it establishes a one-to-one correspondence between the perspectives of the learners and possible outcomes. By using a bipartite graph, the edge orientation problem in a one-inclusion graph can be effectively transformed into a bipartite matching problem. Figure 4 shows such a bipartite graph for the transductive model adapted from the analogy of the hat problem.

Like before, if due to constraints in the realizable setting, a node from $V'$ only has one arc extending from it, then the learner can be certain to label that one as "valid." Otherwise, if no clear node exists, further methods from the learner's prediction strategy will be used to predict it. Using the hat problem as an analogy in bipartite matching has effectively transformed the transductive model into a combinatorial problem.

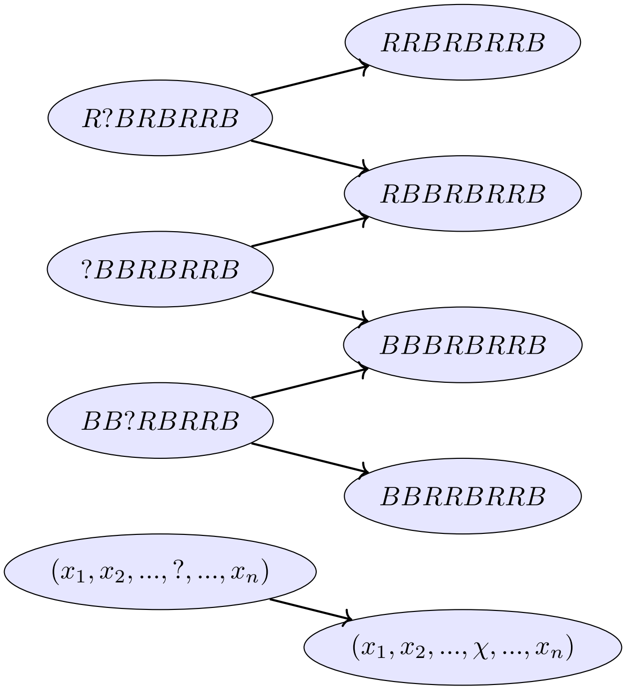


Figure 3: Bipartite graph for hat problem

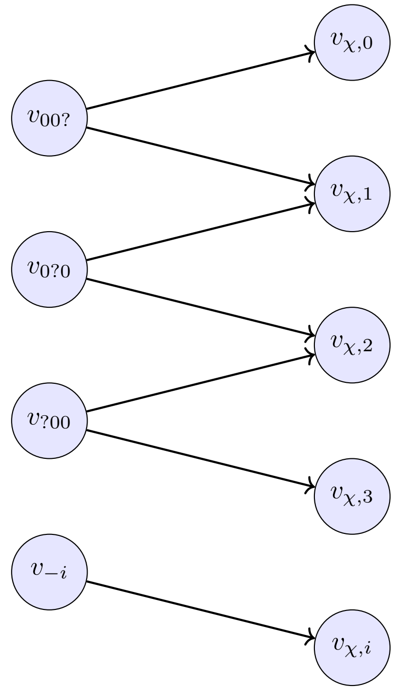


Figure 4: Bipartite graph for the transductive model

## 4.2 Limitations

Because the hat problem and the transductive model still have fundamental differences in some ways, it limits the current potential of applying the full analogy. Some of the key differences between them will be outlined here, along with the limitations involved in connecting the hat problem and the transductive model.

### 4.2.1 Goals

The transductive learning model and the hat problem have different goals to achieve for the players and the learner. Ebert's version of the hat problem (and most versions) aims to maximize the probability of satisfying a "winning" condition, such as guessing at least one label correctly with no mistakes. On the other hand, the success condition for each case in the transductive model is to predict the missing label correctly. Since the dataset in the transductive model is chosen by an adversary, the expected loss would be averaged from the loss in the worst-case scenario. Specifically, it is given by

$$\frac{1}{n} max_{h \in H} \sum l(h(x_i), y_i).$$

This means that to minimize the worst-case loss, the goal of the prediction strategy would be, as emphasized by Dughmi, to "spread out the error".

Most variations of the hat problem differ from this because the goal of the player is to maximize the number of instances where the winning condition is satisfied, not the loss in each instance itself [8]. In some versions of the hat problem, the player is even allowed to skip its turn, further contributing to maximizing the number of winning scenarios. The most important distinction such a problem from the transductive model is that the number of incorrect guesses in each instance does not matter. In the hat problem, the error is defined more discretely. The loss is only checking whether the winning condition is passed or not, regardless of the number of errors or correct guesses. So, the loss can be defined as a 0-1 function where the output is 1 only if there is at least one $i$ such that $h(x_i) = y_i$, and no $i$ such that $h(x_i) \neq y_i$. In Ebert's hat problem, to satisfy the winning condition, the ground truth must have every incident edge pointed towards it or be undirected. Thus, the goal can be equivalently translated to a problem maximizing the vertices with an out-degree of 0, which means that the strategy will exploit the feature of "concentrating the errors together." It is also important to note that, due to this strategy, the optimization of error will only work best for a certain set of dataset sizes, particularly when $n = 2^m - 1$ for any positive integer $m$, which will perfectly clump possible errors into a few cases. Therefore, the goal of the transductive model could be thought of as having the opposite goal of the hat problem. Instead of "clumping the errors" together so that the number of winning scenarios can be maximized, its strategy mainly relies on "spreading out the error" as evenly as possible.

### 4.2.2 Randomization of the dataset

The other important difference that separates the transductive model and the hat problem is how the dataset is chosen. Essentially, the labels for each data point in the hat problem are drawn randomly, whereas the labels are chosen with restriction by a hypothesis space in the transductive model. It is a fairly simple strategy to guess the labels when the data points are randomly chosen, but the strategy becomes less trivial when constraints are placed by a hypothesis space. A particular example of this is seen in the game described by Butler et al. [5], which aims to maximize the guaranteed number of correct guesses in all hat configurations. This is quite similar to the transductive model. However, the optimal strategy in this game only guarantees $\lfloor \frac{n}{m} \rfloor$ correct predictions [5], where $n$ is the number of people and $m$ is the number of colors. This is because the players are given no information on the distribution of hats, which results in the assumption that each hat color are equally likely to occur. However, if restrictions on the pool of hats or how to assign hat colors (e.g. the limited hats game or a hypothesis space) are added, this guarantee can be improved. Essentially, it is easy to get some number of correct answers on average over multiple scenarios, in which case, even a randomized strategy (one where players decide on their guesses based on a dice roll) will suffice [17]. However, when an adversary is involved in the game, the players would need to come up with some strategy that is a function of which position they see each hat.

In addition, in most hat problems, all configurations of hats occur with an equal likelihood. The strategy is thus seen to take advantage of this fact. An example of this is shown in Ebert's version of the hat problem. Because no adversary is always picking the worst-case scenario, each possible hat configuration will appear with equal likelihood. This means that the prediction strategy is able to rely on minimizing the number of worst-case scenarios to increase the winning probability. In fact, for games with $n$ players where $n = 2^m - 1$ [14], the probability of success is $\frac{2^m-1}{2^m}$. A randomly chosen dataset puts no preference for a particular configuration of hats given to the players, which allows them to use such a strategy to minimize the number of instances of failure, instead of minimizing the number of errors for individual instances.

### 4.2.3 Strength of the adversary

Another important distinction between the transductive model and the hat problem lies in the adversary. Previous sections explained versions of the hat problem where hats are randomly drawn from a set of differently-colored hats. In some versions of the hat problem, though, there does exist an adversary, which makes it more similar to the transductive model. However, there is still a key difference: the adversary has a very different level of power (it could be either weaker or stronger) than in the transductive model. To see this, recall that an adversary in the transductive model is restricted by a certain hypothesis $h \in H$, whereas the adversary in the hat problem is mostly limited by the number of hats of each color. This means that the adversary in the hat problem can choose from many more configurations of datasets in most cases. Empirically, the transductive learning model with $n$ data points can minimize its errors according to its VC dimension, specifically $\Pr[y_i \neq f(x_i)] \leq \frac{d}{n+1}$, where $d$ is the VC dimension [16]. However, in the hat problem, the power of the adversary could have a very different range. For instance, consider an adversary choosing colors from $C = \{c_1, c_2, ...c_m\}$, where $c_i \in C$ is the number of hats of the $i$th color. The adversary could be set to a very weak strength, such as when $c_1 + c_2 + ... + c_m = n$, in which case every player could guess correctly. The adversary could also be very strong, such as when $c_1 = c_2 = ... = c_m = n$, in which case the players can only guarantee $\lfloor \frac{n}{m} \rfloor$ correct guesses [5]. Essentially, the strength of the adversary depends on the number of possible dataset configurations it can choose from. In the hat problem, this number can vary greatly depending on what restrictions are placed, whereas the transductive model's adversary has a more consistent strength.

The strength of the adversaries in the agnostic setting and a hat problem with no restriction can also be compared. In the agnostic setting, although the adversary could choose any arbitrary dataset, the error is evaluated against the benchmark hypothesis class [6]. So, the learner can still make predictions relatively close to the correct labels. However, in the hat problem, there is no information that the players know about the dataset patterns picked by the adversary, which makes it much more difficult to improve the effectiveness of the strategy.

### 4.3 Error comparisons

This section will draw comparisons on how error can be indicated in the hat problem and the transductive model. In both problems, a correct guess is defined as orienting the edge towards the correct node node [4].

The expected loss can be measured in the one-inclusion graph. In the realizable setting, this is seen as the maximum out-degree of a node divided by $n$. In the agnostic setting, finding the error requires finding the Hamming distance from the node with the best possible hypothesis. For some hat problems, particularly problems with an adversary, this is similar — the error rate would be the out-degree of the worst-case scenario divided by $n$ [5]. However, most hat problems do not involve an adversary and still use randomly drawn labels, so their expected losses are calculated as the probability of failing the winning condition. This is calculated as the total nodes with an out-degree of 0 divided by the total number of configurations. Unfortunately, due to the different natures of their conditions, the two problems cannot draw a completely equivalent comparison for their respective errors.

Additionally, by using the hat problem analogy, some error bounds can also be established. Particularly, the maximum correct hats game aims to have players guess as many hats correctly as possible, and the strategy aims to guarantee a certain number of correct guesses in each case. It is possible to find out the lower- and upper-bounds of these correct answers, which will be elaborated more in section 5.

## 5 Future Steps

In this section, further connections between the hat problem variations and the details of the transductive model will be made. This section will also speculate about possible future directions for research in transductive learning that uses the applications from hat problem solutions.

### 5.1 Putting restrictions on the Hamming space

To make the hat problem progressively similar to the realizable setting of the transductive model, it is useful to apply the limited hat variation of the hat problem. This was described previously in the paper as the game in the work of Butler et al., where imposing constraints on the number of hats the adversary can help eliminate "impossible" nodes of hat configurations. In the limited hats game, the players can improve on their maximum guaranteed number of correct guesses. Previously, when there was no restriction, or $c_0 + c_1 + ...c_m = n$ where $c_i$ is the number of hats of the $i$th color, the guaranteed number of correct answers was only $\lfloor \frac{n}{k} \rfloor$. It is easy to see that this guarantee is optimal since each player will guess approximately $1/k$ of the time [5]. The strategy for this depended on a modular arithmetic approach where

With restrictions, however, it is possible to increase this maximum correct guess guarantee if the new information from the restrictions is utilized. For example, if players know that each color has no more than 2 hats in a three-player game, they can improve the 1 guaranteed correct prediction to 2. This strategy is created by the players forming a closed cycle with each player guessing the opposite of the color of the next neighbour [4]. Using the limited hats game variation, many interesting properties could be found and applied to the transductive model. Specifically, the number of guaranteed correct answers can provide valuable insights into the minimum guaranteed correct predictions in the transductive model. Since having a hypothesis class in the transductive model allows for more correct predictions than no restrictions in the hat problem, the error in the transductive model cannot exceed this number in the hat problem. Thus, one potential research direction is to consider how the connection between limited hat games and transductive models can give a useful estimation of the maximum error of transductive models.

## 5.2 Edge orientation methods

Edge orientation techniques from hat problems could potentially be applied to derive alternative ways to orient the edges in the one-inclusion graph. One should start by ensuring that the edge orientation strategy in the hat problem achieves the opposite effect as the transductive learning model. That is, the hat puzzle should aim to concentrate errors, and codewords will have an in-degree of 0 and an out-degree of $n$. Conversely, edge orientation on the transductive model should aim to spread out in- and out-degrees evenly.

Furthermore, another strategy from the hat problem could be applied to orient the edges on the OIG. Employing an Eulerian circuit is often a solution to guaranteeing a certain number of correct guesses in adversarial conditions of the hat problem [5]. For example, this can be seen in the 2-player case of the hat problem: it can always guarantee one correct guess using this strategy. The first person would guess the opposite of the second color, and the second person would guess the same as the first color, creating an Eulerian circuit of 4 vertices, which is shown in Figure 5.

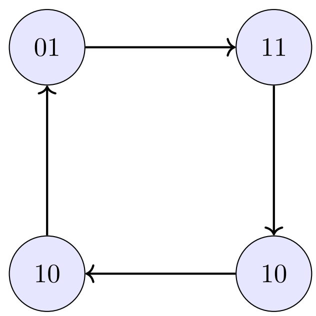


Figure 5: An Eulerian circuit of 4 vertices

Players can simply pair up and employ the same strategy for more people. According to Butler et al., the optimal strategy guaranteeing $\lfloor \frac{n}{k} \rfloor$ correct guesses occurs when there is an Eulerian circuit in the Hamming graph [5], which corresponds to when each vertex of the hypercube has an equal in-degree as the out-degree. Although these strategies cannot immediately guarantee the optimal solution, they can still offer a clever way to deduce the set of best edge orientations. Therefore, the question for future research is: how might using Eulerian circuits and the Hamming graph be applied to cleverly search for edge orientations in the one-inclusion graph?

## 5.3 Hat problems with an adversarial allocation over a known pool of hats

So far, except for Butler's hat problem, only hat problems that involve a randomly drawn set of hats have been discussed, which rely on strategies that depend on an equal probability of getting each ground truth. Another version of the hat problem specified in the paper of Feige et al. [17] that is more directly connected to the transductive model will be described in the following two subsections.

The basis of Feige's hat problem lies in building robust strategies in the face of an adversary. Consider a hat puzzle with an adversary where the players try to maximize their number of correct guesses in a single instance. Strategies commonly used in the hat problem that depends on a randomized dataset might guarantee some number of correct guesses on average but could guarantee no correct answers when an adversary is present. It is thus necessary to consider the worst-case scenario when formulating these strategies. This variation of the puzzle forces the players to ultimately be effective against every possible dataset that the adversary chooses, achieved through a non-symmetric strategy argued by Feige et al. [17].

In this variation of the puzzle, the players utilize known information about the restrictions on the pool of hats to assist in their strategy. This can be compared to the set of ground truths restricted by a hypothesis class. However, it is important to note that unlike the transductive model, where the learner can infer information on the permutation of hats, the hat problems discussed only provide information on the number of hats of each color.

Contrary to the strategy used in most variations of the hat problem, this section will discuss the strategy known as the symmetric strategy. This is when the players' strategy is a function of how many hats of each color they see. According to Feige et al. [17], a symmetric strategy will often be susceptible to exploitation by the adversary, because an adversary can easily arrange a scenario where large numbers of people see the same number of hats of each color and guess incorrectly. Thus, a non-symmetric strategy was introduced, in which players make predictions not only as a function of the number of hats in their perspective but also as a function of how they are arranged. This section will illustrate this with the "discarded hat game," which is also described in their paper [17]. In this game, an adversary assigns $4k-1$ hats on the players' heads. The players will have previous knowledge of the pool of possible configurations: there are $2k-1$ hats of one color (the minority color) and $2k$ hats of another (the minority color). However, no information is given about which hat color is the majority or the minority. Notice that if a player's hat color belongs to the minority, he will see $2k-2$ hats of his own color and $2k$ hats of the other color, so he can easily predict his own hat color. However, the majority color players will see $2k-1$ hats of each color and realize that he is in the majority, but he will not know which color is the majority color, meaning that he cannot predict their own hat color with certainty. At this point, using non-symmetric strategies can allow the majority-color players to guess correctly $k$ out of the $2k$ times [17].

The non-symmetric strategy introduced is as follows: for the players wearing majority-colored hats, he would only consider the $2k-1$ following players clockwise. The player would then make a prediction of his own hat color in agreement with the color that he sees the most among these $2k-1$ hats. The result of this strategy is that no matter what dataset the adversary chooses, it can always guarantee $3k-1$ of the players guess correctly.

Given that this variation of the hat problem is of a very similar nature to the transductive model, a closer connection can be drawn to it. Especially, this illustrates the importance of using non-symmetric strategies in the transductive model. When the transductive learner is facing a scenario of ambiguity, it is often useful to predict by considering the position of the unlabelled data points, instead of the number of data points in each class.

### 5.3.1 A generalized problem

The discarded hat problem can be generalized to a version with any number of players $n$ and any set of colors $C$ with $|C| \geq 2$, making an analogy to the transductive model more useful. In the generalized variation, the players do not need the pool of hats to have $2k$ of one color and $2k-1$ of the other, but can instead play with any number of each hat. The strategy works similarly to the basic version, based on the principle of distributing the correct guesses among different players with the same colored hats in each scenario (the strategy will be elaborated in the next subsection). In the transductive model, the generalized variation of the discarded hat problem corresponds to a transductive model that can guarantee a number of correct guesses in all scenarios, for any number of classes, with each class having any arbitrary number of labels. It is proven that there exists a strategy that can be used so that the number of guaranteed correct guesses for each color group $H_j$ is between $\lfloor \frac{|H_j|}{|C|} \rfloor$ and $\lceil \frac{|H_j|}{|C|} \rceil$ [17].

Given this variation of the hat problems and the perspective on the minimum guaranteed answers, it is also useful when describing the minimum accuracy under adversarial conditions in the transductive model.

## 5.4 Fractional strategies

The main strategy described for the generalized version of the discarded hat problem is a fractional strategy. This strategy is based on confidence levels that the players will have about their predictions. For example, a learner facing a certain masked dataset may conclude, based on known restrictions, that one scenario is more plausible than another. Therefore, the learner can assign a probability or confidence level to a label. In the hat problem, this can be mimicked through fractional strategies. In the strategy, a player does not assign a single, decisive choice for a prediction but instead assigns a probability to a node in the possible configurations (recall that these are nodes from the set $v$ in the bipartite graph). A fractional strategy used in the hat problem is a mapping $s_i : C^{n-1} \to C$ such that it assigns a probability label $z_{i,\chi,h_i}$ for every possible guess for player $p_i$ [17]. Applying this to the transductive model, for each dataset shown to the learner $v_{-i}$, it would have a probability $z_{i,\chi,h_i}$ that the learner at position $i$ guesses $\chi$ when shown the masked dataset $h_i$.

This section will connect the above strategy to fractional strategies in the transductive learning model. To implement this strategy, it will again involve the usage of the bipartite graph in [4,1]. For each node $v_{-i}$ shown to the learner, the arc connecting it to a ground truth node $v_{\chi,i}$ would be assigned a probabilistic label $z_{i,\chi,h_i}$ that represents the chance of the learner predicting the missing label as $\chi$. The fractional strategy satisfies two conditions: 1) $z_{i,\chi,h_i}$ is always greater or equal to 0, and 2) $\sum_{g\in C} z_{i,\chi,h_i} = 1$ [17].

These conditions are fairly easy to see — the probability of guessing any label will always be non-negative and the probabilities of guessing each of the colors when seeing a dataset $h_i$ will add up to 1. The bipartite graph also needs to satisfy the additional condition that the number of all the probability labels on the guesses is within the target fraction of correct predictions that the learner wants to guarantee. In other words,

$$\lfloor \frac{|H_j|}{|C|} \rfloor \geq \sum_{i\in H} z_{i,g,h_i} \leq \lceil \frac{|H_j|}{|C|} \rceil.$$

Then, the learner takes a rounding procedure that rounds all the probability labels to the nearest integer, under which all of the above conditions are invariant [17]. So, at the end of the rounding procedure, each of the labels on the arcs will be either 0 or 1. When all arcs are labelled with a value from $\{0, 1\}$, it indicates each bipartite edge as "valid" or "non-valid", and produces a bipartite matching for our strategy.

The fractional strategy allows the learner to successfully find a bipartite matching from probabilities assigned to the label predictions. This strategy works to distribute the correct guesses evenly among the different groups of players by making sure that they always satisfy the goal constraint (the target number of correct guesses), $\lfloor \frac{|H_j|}{|C|} \rfloor \geq \sum_{i\in H} z_{i,g,h_i} \leq \lceil \frac{|H_j|}{|C|} \rceil$, which was previously discussed to be an optimal strategy.

## 5.5 Sight graphs and multiple unlabelled data points

This section pin points out the possible applications of hat problems where players have a limited view of other players. This variation of the hat problem allows the transductive model's analogy to be extended to datasets where there are various unlabelled data points. To turn a single-prediction model to have many unlabelled data points, sight graphs in the hat problem can be used to simulate it. Sight graphs are graphical representations of the data points viewable by each player. In this version of the hat game, to represent players who are not able to be seen by another player, the problem utilizes both a guessing graph $G_g$ and a sight graph $G_s$ [5]. In the sight graph, nodes are players in the hat game, or equivalently, a particular position $i$ in a masked dataset $v_{-i}$. A node $v_i$ in $G_s$ has a directed edge towards $v_j$ if player $v_i$ can see $v_j$, otherwise, there would be no edge between them. If two players have mutual vision of each other, such as in all the hat problems previously discussed in this paper, the edge between the two nodes is directed toward both directions.

In the transductive model, there is a set of labelled data points and some unlabelled data points. The learner can see all the labelled data points, but from no perspective will the learner be able to see any of the unlabelled ones. Thus, the unlabelled data points are represented as nodes where no other node $v_{-i}$ has vision on it. However, those nodes would still have directed edges toward the other labelled data points in the training set, since from their perspectives, the points from the training dataset are visible. For example, the sight graph in Figure 6 shows a data set with points $\{a, b, c, d, e\}$, where $\{a, b, c\}$ are the labelled data points and $\{d, e\}$ are the unlabelled data points. Except for nodes $d$ and $e$, there is an edge connecting every pair of nodes. Directed edges extend from $d$ and $e$ to $a$, $b$, and $c$. Additionally, each edge between any two points from $\{a, b, c\}$ is undirected to represent mutual visibility [5].

Based on theorems for the sight graph proven by Butler et al. [5], one can find the lower bound and the upper bound for the minimum guaranteed correct predictions for a transductive learning model for multiple unlabelled data points. A key theorem [5] states that if $c(G)$ is the maximum number of vertex disjoint cycles, and $G_s$ must remove at least $F(G)$ nodes to be an acyclic graph, then the maximum number of correct predictions a strategy can guarantee $H(G)$ follows $c(G) \leq H(G) \leq F(G)$. Therefore, for our example data set $\{a, b, c, d, e\}$, one can establish bounds on the correct answer guarantees. Specifically, the number of guaranteed correct answers would have a lower bound of 1, since $\{a, b, c\}$ form a vertex disjoint cycle. That number would have an upper bound of 2 because removing any two nodes from $\{a, b, c\}$ will turn the graph into an acyclic one.

By using sight graphs, transductive models with multiple unlabelled data points can be represented in an alternative way. This both gives a new approach to interpreting this problem and gives new ways

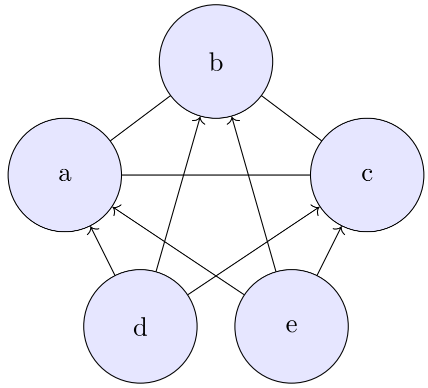


Figure 6: A sight graph for 2 unlabelled data points

to find bounds for estimating the error. More research in the future could be done with transductive learners with multiple unlabelled data points. Potentially, it could employ this approach of a sight graph in the hat problem, which allows one to treat multiple unlabelled data point problems in the same way as the transductive model with a single prediction.

## 6 Conclusion

Transductive learning is gaining popularity as a field of active research. It notably allows models to skip the intermediary step of inferring a function before predicting on datasets. While it has previously been compared to PAC learning models, it has not yet made a connection to other concepts. This paper makes a connection between the transductive model and the hat problem, a classic combinatorial puzzle in mathematics. Because of their highly analogous nature, doing so could yield many useful insights into transductive learning. This paper, by formally analyzing the connections between the hat problem and transductive learning, proffers the reader the idea that the transductive learning model is, in many ways, almost equivalent to the hat puzzle.

Similar to how the hat problem uses prediction strategies as a function of other known hats to guess a missing hat color, the transductive learning model also focuses on directly making predictions for a dataset by considering the labelled data points. This paper involved a technical analysis regarding the similarities between the two problems' setups, providing a mathematical framework to understand the two. Additionally, it pointed out some of the fundamental distinctions in the nature of their setups that prevented a full analogy.

A further technical analysis brought about the connections between the graphical representations of the two problems, comparing the Hamming graph with the one-inclusion graph, which also allowed for a comparison between the two problems' error measurements. It then converted the graphical representations of the transductive model into a bipartite matching problem, using the hat problem as a medium for interpreting the bipartite graph. This paper also compared these representations under different settings (agnostic and transductive) of the transductive learning model.

Furthermore, the analysis involved extensions with many versions of the hat problem. These variations expanded the transductive model to include multiclass classification and several unlabelled data points. Variations of the hat problem were also discussed further to establish a closer connection to the transductive learning model, especially in the realizable setting. This was brought about using adversarial hat problems. These various connections were introduced along with proposing their potential applicability to further research in transductive learning.

Overall, analyses through the perspective of the hat problem are important to understanding the newly developing area of transductive learning and offer a fresh approach. Through the many analogies with the hat problem, the paper suggests insights that could be beneficial to advancing transductive learning research in the future.

## Acknowledgements

I thank my professor, Dr. Shaddin Dughmi, for engaging in interesting and helpful discussions about the topic throughout much of this research paper. Most importantly, he pointed out during our conversations the possible connections to transductive learning of the hat problem.

## References

[1] Aggarwal, G., Fiat, A., Goldberg, A. V., Hartline, J. D., Immorlica, N., & Sudan, M. (2011). Derandomization of auctions. *Games and Economic Behavior*, 72(1), 1–11. https://doi.org/10.1016/j.geb.2010.07.007

[2] Alon, N., Omri Ben-Eliezer, Shangguan, C., & Itzhak Tamo. (2020). The hat guessing number of graphs. *Journal of Combinatorial Theory Series B*, *144*, 119–149. https://doi.org/10.1016/j.jctb.2020.01.003

[3] Asilis, J., Devic, S., Dughmi, S., Sharan, V., & Teng, S.-H. (2024). *Regularization and Optimal Multiclass Learning*. https://arxiv.org/pdf/2309.13692

[4] L. Bushi, J. (2012). *OPTIMAL STRATEGIES FOR HAT GAMES*. https://web.pdx.edu/~caughman/Jaime501.pdf

[5] Butler, S., Hajiaghayi, M. T., Kleinberg, R. D., & Leighton, T. (2008). Hat Guessing Games. *SIAM Journal on Discrete Mathematics*, 22(2), 592–605. https://doi.org/10.1137/060652774

[6] Dughmi, S., Kalayci, Y., & York, G. (2024). *Is Transductive Learning Equivalent to PAC Learning?* https://arxiv.org/pdf/2405.05190

[7] Dylan, B., & Saint, J. (n.d.). *The Hat Problem*. https://mathstat.slu.edu/~johnson/public/maths/hatproblem.pdf

[8] Guo, W., Kasala, S., Rao, M., & Tucker, B. (n.d.). *The Hat Problem And Some Variations*. https://web.njit.edu/~wguo/Hat%20Probelm.

[9] Haussler, D., N. Littlestone, & Warmuth, M. K. (1994). Predicting {0, 1}-Functions on Randomly Drawn Points. *Information and Computation*, *115*(2), 248–292. https://doi.org/10.1006/inco.1994.1097

[10] He, X., & Li, R. (2020). Hat Guessing Numbers of Degenerate Graphs. *The Electronic Journal of Combinatorics*, *27*(3). https://doi.org/10.37236/9449

[11] Liu, H., & Yang, Y. (2015). *Bipartite Edge Prediction via Transductive Learning over Product Graphs*. https://proceedings.mlr.press/v37/liuc15.pdf

[12] M. Farnik, A hat guessing game. PhD thesis, Jagiellonian University, 2015.

[13] Numberphile. (2020). Hat Problems - Numberphile [YouTube Video]. In *YouTube*. https://www.youtube.com/watch?v=laAtv310pyk

[14] Slater, M. (2022a, March 29). *Hamming Hats*. Thenumb.at. https://thenumb.at/Hamming-Hats/

[15] T. Ebert, *Applications of recursive operators to randomness and complexity*, Ph.D. Thesis, University of California at Santa Barbara, 1998.

[16] The Learning Theory Alliance Blog. (2024). *One-Inclusion Graphs and the Optimal Sample Complexity of PAC Learning: Part 1 – The Learning Theory Alliance Blog*. Let-All.com. https://www.let-all.com/blog/2024/06/06/one-inclusion-graphs-and-the-optimal-sample-complexity-of-pac-learning-part-1/

[17] U. Feige, *You can leave your hat on (if you guess its color)*, Technical Report MCS04- 03, Computer Science and Applied Mathematics, The Weizmann Institute of Science, 2004.

[18] Wikipedia Contributors. (2024, August 9). *Transduction (machine learning).* Wikipedia; Wikimedia Foundation. https://en.wikipedia.org/wiki/Transduction_(machine_learning)

[19] Zou, A. W., & Tigges, P. (2024). *Report on Hat Guessing Games by Steve Butler, Mohammad T. Hajiaghayi, Robert D. Kleinberg, and Tom Leighton*. Uiowa.edu. https://homepage.math.uiowa.edu/~idarcy/COURSES/151/SPRING09/Report_AngWeiZou_PeterTigges.doc