Unifying Predictive Multiplicity of Classification and Link Prediction

Jonathan Schnitzler[1] and Lukas Harsch[1]

University of Stuttgart, Stuttgart 70174 BW, Germany

Abstract. In machine learning, models are designed to detect patterns from data, enabling tasks like image recognition and financial forecasting. Despite similar performance metrics, different models can yield conflicting predictions, a phenomenon known as multiplicity. This inconsistency, particularly in critical applications like recidivism prediction raises concerns about reliability and fairness. Building on foundational work, including Breiman's focus on interpretability and recent studies by Marx et al. and Zhu et al., this paper explores predictive multiplicity, consolidating definitions into a unified framework, providing illustrative examples, and proposing mitigation strategies. Our analysis aims to deepen understanding and guide future research in this emerging area.

Keywords: Predictive Multiplicity · Classification · Knowledge Graph Embeddings

1 Introduction

In the field of machine learning, the objective is to develop algorithms and models capable of learning patterns and relationships directly from data, without requiring explicit programming. These models are subsequently applied to tasks such as making predictions or informing decisions. For instance, machine learning can be utilized to identify objects within an image or to determine the optimal timing for stock transactions.

The training process for such models involves minimizing a predefined error metric, with the aim of achieving high accuracy when the model is applied to previously unseen data. Notably, it has been observed that, for the same problem and dataset, multiple models may emerge as viable solutions. These competing models—each employing different architectures but trained on identical data—often exhibit comparable performance (e.g., accuracy) when evaluated on unseen data. This phenomenon is commonly referred to as **multiplicity.**

The implications of model multiplicity have historically been underappreciated and primarily regarded as a challenge in model selection, necessitating trade-offs among computational efficiency, interpretability, and accuracy [1]. However in 2001, Breiman's work [2] marked a pivotal shift in perspective by emphasizing the interpretability of explanations provided by individual models. He highlighted that if multiple competing models yield differing explanations for the same result, it raises critical concerns about the reliability and consistency of such interpretations.

Building on this foundation, Marx et al. [3] introduced the concept of **predictive multiplicity**, which arises when competing models produce conflicting predictions for individual data points. In their study, they trained multiple competing models on the ProPublica COMPAS dataset for the task of recidivism prediction. They found that a model with just 1% lower accuracy than the most accurate model assigned conflicting predictions to over 17% of individuals in the dataset. Additionally, they observed that the predictions for 44% of individuals were affected by the model choice.

These findings underscore the significant real-world implications of predictive multiplicity, particularly in high-stakes applications. For example, tools such as PATTERN [4] are already in use in the United States to evaluate the likelihood of recidivism among incarcerated individuals, where a positive assessment can result in early release. The potential for conflicting predictions highlights the critical need to address predictive multiplicity to ensure fairness and reliability in such systems.

In a more recent study, Zhu et al. [5] demonstrated that predictive multiplicity can arise when utilizing different embedding methods for knowledge graphs (KGs) in link prediction tasks. Their analysis assessed predictive multiplicity across six representative knowledge graph embedding (KGE) methods on widely used benchmark datasets. The study identified significant predictive multiplicity in link prediction, with conflicting predictions observed in 8% to 39% of testing queries, highlighting the impact of this phenomenon within the domain of knowledge graph embeddings.
To address predictive multiplicity, the authors applied voting methods inspired by social choice theory and demonstrated their effectiveness in mitigating this issue.

As predictive multiplicity is a relatively new and underexplored topic, our aim is to provide a thorough summary of the current state of research while offering explicit examples to establish a solid and comprehensive foundation for future investigations. To achieve this, our report includes the following key components:

1. **Foundational Concepts:** We outline the necessary foundations for understanding predictive multiplicity in both classification tasks and link prediction tasks involving knowledge graph embeddings (KGEs).
2. **Framework Development and Application:** We consolidate the formal definitions presented by [3] and [5] into a comprehensive framework that characterizes predictive multiplicity in a task-agnostic manner. This framework is subsequently applied to illustrative examples to demonstrate its utility and broad applicability.

3. **Mitigation Strategies:** We discuss how methods derived from social choice theory can be applied to address and alleviate predictive multiplicity effectively.
4. **Causal Analysis:** We analyze the underlying reasons why predictive multiplicity occurs across different KGE methods, shedding light on the factors contributing to this phenomenon.

2 Foundations

This section outlines the fundamental principles essential for comprehending classification tasks and link prediction in knowledge graphs utilizing Knowledge Graph Embeddings (KGEs). The subsequent examples are grounded in these foundations to ensure clarity and coherence. For classification, we adopt the definitions articulated by [3], while for knowledge graphs, we rely on the definitions provided by [5].

2.1 Classification

We formally define a dataset of n examples $\mathcal{D}_{\text{train}} = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n}$ where each example consists of a feature vector $\mathbf{x}_i = (1, x_{i1}, x_{i2}, \ldots, x_{id}) \in \mathbb{R}^{d+1}$ and a label $y_i \in \{\pm 1\}$. This dataset is then used to fit a classifier $h : \mathbb{R}^{d+1} \to \{\pm 1\}$ from a hypothesis class $\mathcal{H}$ by maximizing the accuracy:

$$h \in \operatorname*{argmax}_{h \in \mathcal{H}} \hat{A}(h) \tag{1}$$

with the accuracy

$$\hat{A}(h) := \frac{1}{n} \sum_{i=1}^{n} \mathbb{1}[h(\mathbf{x}_i) = y_i]. \tag{2}$$

The $\mathbb{1}$ operator assigns a value of 0 to misclassified instances and 1 to correctly classified instances. Consequently, an optimal classifier would achieve an accuracy of $\hat{A}(h_{\text{optimal}}) = 1$.

2.2 Knowledge Graphs

Knowledge graphs (KGs) structure data in a graph-based format, comprising entities (such as people, places, or concepts) represented as nodes, and relationships between them depicted as edges (links). They serve to organize and integrate data from diverse sources, facilitating advanced data analysis and reasoning by encapsulating the semantic relationships and contextual information inherent in the data.

Entities e from a set E and their relationships $r \in R$ are represented in a knowledge graph through a collection of triples $tr = \langle h, r, t \rangle$, where h is the head entity and t is the tail entity. Thus, a KG is a subset of all possible triples, denoted as $\mathcal{G} \subset E \times R \times E$. A knowledge graph embedding (KGE) model assigns a predictive score to each triple, reflecting the plausibility that the triple is valid:

$$M_\theta : E \times R \times E \to \mathbb{R}, \tag{3}$$

where θ represents the model's parameters. The model M_θ is trained to assign higher predictive scores to positive triples (true facts) and lower scores to negative triples (false facts), thereby distinguishing between plausible and implausible relationships.

Link Prediction in knowledge graphs aims to identify new relationships between entities within the graph. In this process, a query $q \in Q$ represents an incomplete triple, where either the head or the tail entity is unknown, denoted as $\langle h, r, ? \rangle$ or $\langle ?, r, t \rangle$. The model M_θ assigns predictive scores to all possible candidate triples formed by substituting the unknown entity with each entity $e \in E$, resulting in triples such as $\langle h, r, e \rangle$ or $\langle e, r, t \rangle$.
The entities are then ranked based on their predictive scores, from highest to lowest. The entity with the highest score is selected as the most likely correct answer to the query, and the corresponding triple is added to the knowledge graph. The ranking process is defined formally as:

$$e_1 \preceq_{M_{\theta,q}} e_2 \iff M_{\theta,q}(tr(q, e_1)) \leq M_{\theta,q}(tr(q, e_2)). \tag{4}$$

where the order relation $\preceq_{M_{\theta,q}}$ indicates that entity e_1 precedes or is equal to entity e_2 in the ranking based on the scores assigned by M_θ. For a finite set of entities, the rank of an entity is determined by counting the number of entities that precede it in the ordering:

$$R_{\preceq_{M_{\theta,q}}}(e) = |\{d \in E | e \preceq_{M_{\theta,q}} d\}|.^1 \tag{5}$$

Link prediction has the unique characteristic that, for a given query, there may be multiple valid entities e. For instance, in a social network, a person (head) may be friends (relation) with multiple individuals (tails). In such cases, the ranking of all correct tail entities is arbitrary and does not provide additional useful information. Therefore, it is common to evaluate the top K predictions made by the model M_θ for a query to evaluate the models performance. If the correct entity for a sample appears within the top K predictions, we consider the model's prediction to be correct. Formally, the top K predictions can be defined as follows:

[1] We note that this definition differs by $+1$ compared to Eq. (3) in [5]. This adjustment was made because, under the original definition, each entity is compared to itself, causing the $+1$ to be automatically included for all entities. Therefore, there is no need to add it explicitly.

$$T_K(M_\theta, tr(q,e)) = \mathbb{1}[R_{\preceq_{M_{\theta,q}}}(e) \leq K]. \tag{6}$$

where $\mathbb{1}$ is the indicator function. Based on this definition, the accuracy of the model M_θ is referred to as Hits@K, and is calculated as:

$$H_K(M_\theta) = \frac{1}{n} \sum_{(q,e)\in\mathcal{T}} T_K(M_\theta, tr(q,e)) \tag{7}$$

where $\mathcal{T}$ is the test set of size $|\mathcal{T}| = n$.

2.3 Knowledge Graphs Embeddings

This section is based on Ge (2023) [7].
Knowledge Graph Embeddings (KGEs) can generally be categorized into distance-based and semantic-based approaches. Distance-based KGEs model relationships as translations in a continuous vector space. A common example of a distance-based KGE is TransE. Let $\mathbf{h}, \mathbf{r}, \mathbf{t} \in \mathbb{R}^d$ denote the embeddings for the head, relation, and tail of a triple, respectively. The scoring function for TransE is defined as:

$$f_r(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \|\mathbf{h} + \mathbf{r} - \mathbf{t}\|_p \tag{8}$$

where $p = 1$ or 2 denotes the 1-Norm or 2-Norm, respectively. In this formulation, the sum of the head embedding $\mathbf{h}$ and the relation embedding $\mathbf{r}$ should approximate the tail embedding $\mathbf{t}$.
Semantic-based KGEs represent the relationships between head and tail entities as matrices rather than vectors, to measure a semantic matching score. The goal is to capture richer interactions between entities by considering pairwise interactions in matrix form. A common example is the RESCAL KGE. Given $\mathbf{h}, \mathbf{t} \in \mathbb{R}^d$ and $\mathbf{M}_r \in \mathbb{R}^{d\times d}$, the scoring function is given by:

$$f_r(\mathbf{h}, \mathbf{r}, \mathbf{t}) = \mathbf{h}^T \mathbf{M}_r \mathbf{t}. \tag{9}$$

Distance-based KGEs are simple and efficient, performing well for relationships where a linear transformation can capture the relation. However, they struggle with more complex relations, such as one-to-many, many-to-one, or many-to-many. Semantic-based approaches, on the other hand, can model complex relationships more effectively by capturing the interactions between entities more comprehensively, but are more computationally intensive due to the need to learn and store large relation matrices, which can lead to overfitting on small datasets.

3 Predictive Multiplicity

Classification and link prediction exhibits predictive multiplicity if competing models $h \in \mathcal{H}$ assign conflicting predictions, even though their general performance is comparable.

3.1 Notation

To unify the definitions of predictive multiplicity for classification from Marx [3] and link prediction from Zhu [5], all KGE models $M_\theta \in \mathcal{M}$ will be denoted as $h \in \mathcal{H}$ and the baseline model M_θ^* will be denoted as h_0. The generalization of $\hat{A}(h)$ and $H_K(h)$ is a discrete loss which measures the performance of a model by counting how many samples $\tau \in \mathcal{T}$ satisfy a statement $S(h, \tau)$,

$$\mathcal{L}(h) = \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \mathbb{1}[S(h, \tau)] \tag{10}$$

for a model $h \in \mathcal{H}$.

For **classification**, the set $\mathcal{T}$ contains feature vectors and labels $\tau = (\mathbf{x}, y)$. The statement $S(\cdot, \cdot)$, corresponds to accuracy $\hat{A}(\cdot)$ from Eq. 2

$$\mathbb{1}[S(h, \tau)] = \mathbb{1}[h(\mathbf{x}) = y]. \tag{11}$$

For **link prediction**, the set $\mathcal{T}$ contains queries and correct answers $\tau = (q, e)$. The statement $S(\cdot, \cdot)$ is whether a triple is in the top-K predictions or not

$$\mathbb{1}[S(h, \tau)] = \mathbb{1}[T_K(h, \tau)] = \mathbb{1}[R_{\preceq_{M_{\theta,q}}}(e) \leq K]. \tag{12}$$

It holds $\mathcal{L}(h) \in [0, 1]$ with the optimal value $\mathcal{L}(h) = 1$ corresponding to perfect accuracy. For convenience define the difference

$$D(h, h_0) := \mathcal{L}(h_0) - \mathcal{L}(h). \tag{13}$$

and for statements $S(\cdot, \cdot)$,

$$\Delta(h, \tau) := \mathbb{1}[S(h, \tau) \neq S(h_0, \tau)] \tag{14}$$

which is indicating deviating predictions of h and the baseline h_0, i.e. multiplicity.

3.2 Definitions

With the established notation it is now possible to define predictive multiplicity as a generalization of Marx [3] and Zhu [5]. Predictive multiplicity is measured on a set of models that perform almost as well as the baseline model.

ε-level Set: Given a a baseline model h_0 and a hypothesis class $\mathcal{H}$ the ε-level set around h_0 is the set of all classifiers $h \in \mathcal{H}$ with a distance of at most ε

$$S_\varepsilon(h_0) := \{h \in \mathcal{H} : D(h_0, h) \leq \varepsilon\}. \tag{15}$$

The set $\mathcal{T}$ is the training or validation set, and thereby $\mathcal{L}(\cdot)$ is the discrete training error. Predictive multiplicity can occur over ε-level sets, where $\varepsilon = 0$. However, in practice ε is typically set to a value greater than zero, as higher training error does not necessarily indicate poorer performance on the test set.

Predictive Multiplicity: Given a baseline classifier h_0 and an error tolerance ε, a prediction problem exhibits predictive multiplicity over the ε-level set $S_\varepsilon(h_0)$ if there exists a model $h \in S_\varepsilon(h_0)$ such that $\Delta(h, \tau)$ is true for some $\tau \in \mathcal{T}$. Predictive multiplicity can be measured via ambiguity and discrepancy.

Ambiguity: The ambiguity of a prediction problem over the ε-level set $S_\varepsilon(h_0)$ is the proportion of points in a set $\mathcal{T}$ that can be assigned a conflicting prediction by a competing classifier $h \in S_\varepsilon(h_0)$

$$\alpha_\varepsilon(h_0) := \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \max_{h \in S_\varepsilon(h_0)} \Delta(h, \tau) \tag{16}$$

Discrepancy: The discrepancy of a prediction problem over the ε-level set $S_\varepsilon(h_0)$ is the maximum proportion of conflicting predictions between the baseline classifier h_0 and a competing classifier $h \in S_\varepsilon(h_0)$:

$$\delta_\varepsilon(h_0) := \max_{h \in S_\varepsilon(h_0)} \frac{1}{|\mathcal{T}|} \sum_{\tau \in \mathcal{T}} \Delta(h, \tau) \tag{17}$$

Ambiguity and discrepency therefore only differ by the position of the maximization. Using basic rules of inequality, the first result can be noted.

Proposition 1. *Ambiguity is always greater than discrepancy.*

$$\forall\, \varepsilon \geq 0 \quad \delta_\varepsilon(h_0) \leq \alpha_\varepsilon(h_0) \tag{18}$$

Furthermore, a bound on the discrepancy can be proven [5].

Proposition 2. *Discrepancy is bounded by the accuracy and ε.*

$$\delta_\varepsilon(h_0) \leq 2 \cdot (1 - \mathcal{L}(h_0)) + \varepsilon \tag{19}$$

Noteworthy, the distance of the accuracy to 1 corresponds to the ratio of opposite predictions, refered to as empirical risk [3].

3.3 Example Classification

Consider a detection algorithm designed to identify potential violent behavior between two individuals at a football stadium, where x_i indicates which team each person supports. This resembles an exclusive-or gate (XOR), which is depicted in Fig. 1. The training data $\mathcal{T}_{\text{train}}$ is sampled as a regular grid with $n = 100$, where the background color additionally indicates the ground truth. The decision boundary (DB) of a baseline $h_0 \in \mathcal{H}$ is depicted as a black dashed line, i.e. every element above the DB is classified as 1, and below as -1. The performance of h_0 is poor with $\mathcal{L}(h_0) = \hat{A}(h_0) = \frac{50}{100} = 0.5$. Three representatives $h \in S_\varepsilon(h_0)$ for $\varepsilon = 0$, i.e. $\mathcal{L}(h) = 0.5$, are considered. A ring-shaped glyph visualizes the predictions on $\mathcal{T} \subset \mathcal{T}_{\text{train}}$ of the baseline classifier h_0 (inner circle) and the discrete epsilon set $S_\varepsilon(h_0)$ (outer ring).

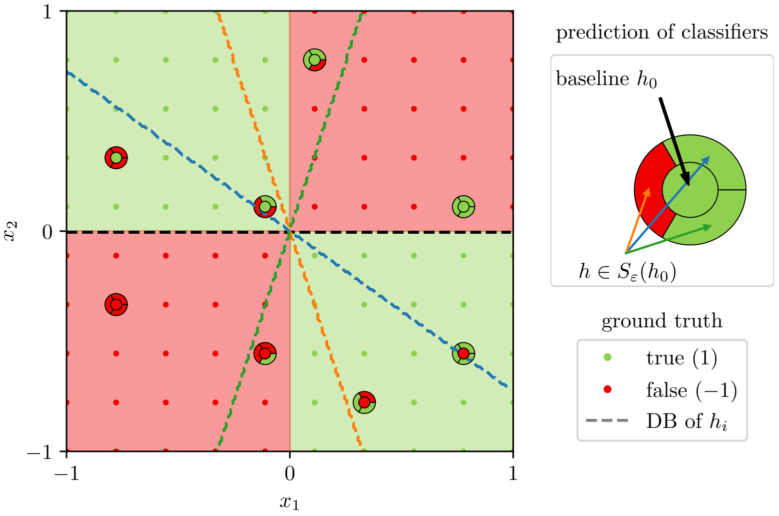


Fig. 1: Classification of XOR dataset with decision bounary (DB) of a baseline classifier h_0 and 3 representatives of $S_\varepsilon(h_0)$ for $\varepsilon = 0$. The prediction of these on the set $\mathcal{T} \subset \mathcal{T}_{\text{train}}$ is additionally highlighted with a ring-shaped glyph

Ambiguity is easily detected by counting the elements $\tau \in \mathcal{T}$ where multiple prediction are present,

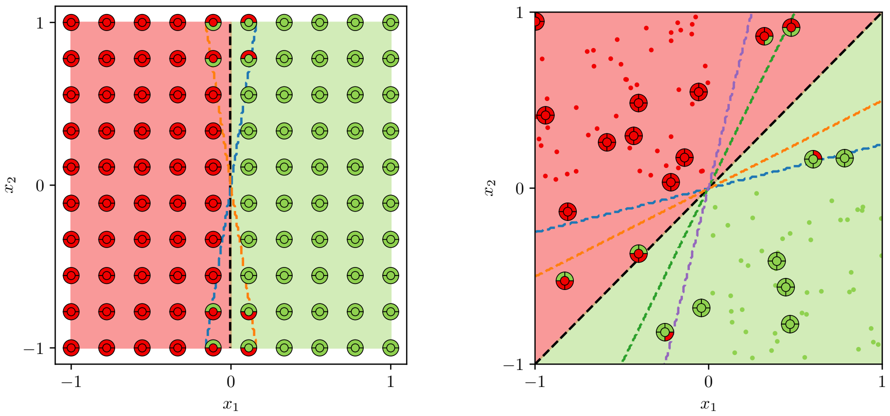
$$\alpha_0(h_0) = \frac{6}{8} \tag{20}$$

To calculate discrepency, for each of the three predictors the mean of $\Delta(h, \tau)$ is calculated individually. Then the maximum, i.e. the classifier with most alternating predictions to h_0, is assigned

$$\delta_0(h_0) = \max\left[\frac{2}{8}, \frac{4}{8}, \frac{6}{8}\right] = \frac{6}{8}. \tag{21}$$

Note, that the different classifiers are generated by rotating the DB around the origin and therefore the green classifier (lower right segment of the glyph) differs the most from the baseline, which leads to equality of ambiguity and discrepancy. Using this, we can observe that the classifier with the exact opposite prediction of the baseline is in the non-discrete set $\tilde{S}_\varepsilon(h_0)$, leading to $\tilde{\delta}_\varepsilon(h_0) = \tilde{\alpha}_\varepsilon(h_0) = 1$.

Two additional classification problems are considered in Fig 2. The bound on $\delta_\varepsilon(h_0)$ of Prop. 2 is illustrated in Fig. 2a. Two classifiers $h_1, h_2 \in S_\varepsilon(h_0)$ for $\varepsilon = 0.05$ are regarded. Since the accuracy $\hat{A}(h_0) = 1$ it holds $\delta_\varepsilon(h_0) < \varepsilon$. The complete trainings data, which is equally sampled as in Fig. 1, is plotted as glyphs.



(a) $\mathcal{L}(h_0) = 1$ bounds discrepancy to $\delta_\varepsilon(h_0) < \varepsilon$, see Prop. 2

(b) $\mathcal{T}_{\text{train}}$ is not covering the space $\mathcal{D}$ leading to PM even for $\varepsilon = 0$ on $\mathcal{T} \not\subset \mathcal{T}_{\text{train}}$

Fig. 2: More examples

Prop. 2 makes no statements for PM on other elements $\tau \in \mathcal{T}$. This is depicted in Figure 2b.

3.4 Example Link Prediction

A knowledge graph regarding the topic of astronomy with two relations will serve as a comprehensive example and is visualized in Fig 3. The same binary glyph as introduced in Fig. 1 is used to indicate whether the entity $e \in E$ for the query

$q = \langle h, r, ? \rangle$ is in the Top-4 prediction, i.e Eq. 6 with $K = 4$. Possible values of baseline model h_0 and two additional models $h_1, h_2 \in S_\varepsilon(h_0)$ are given in Tab. 1. The ground truth is not visualized for brevity and is left as an exercise for the reader. The graph can be understood as a part of a larger dataset like FB15k or WN18 [8]. Consider the query $q = \langle ?, \text{Orbits}, \text{Sun} \rangle$. There is plurality and nuance to the answer. All planets in the solar system orbit the Sun, in this case the entities Mars, Jupiter and Earth are all correct answers. Ordering them could inherit other information like size or closeness to the sun, but in general they are arbitrary. The Moon orbits the Earth, but one could argue that it thereby also orbits the Sun. The same holds for the Hubble Space Telescope.

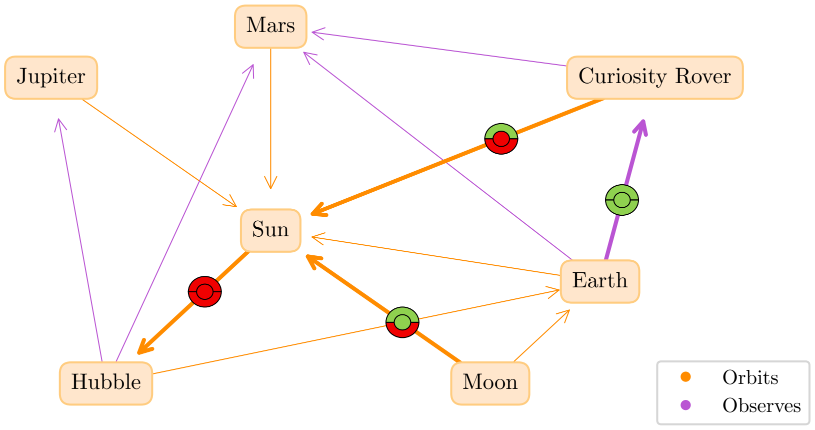


Fig. 3: Astronomy related KG where triplets $\langle h, r, t \rangle$ are visualized as arrows pointing from head entity $h \in E$ to tail entity $t \in e$ with color-coded relation. Predictions are done according to the glyph introduced in Fig. 1

The graph shows some predictive multiplicity with ambiguity $\alpha_\varepsilon(h_0) = \frac{2}{4}$ and discrepancy $\delta_\varepsilon(h_0) = \frac{1}{4}$.

4 Mitigating Predictive Multiplicity through Social Choice Theory

Zhu et al. [5] introduced the application of voting methods from social choice theory as a means to mitigate predictive multiplicity. This section provides a summary of these methods.

Consider a scenario where l competing models $h_1, h_2, \ldots, h_l$ are trained for choosing between m options. In link prediction these are the $m = |E|$ entities for a query q. In classification m is the number of classes, e.g. $m = 2$ for binary classification. For simplicity, we will assume *all against one* voting [6] for the multiclass case.

For each model $h_j(\cdot)$ and prediction $\mathbf{y} \in \mathbb{R}^m$, a scoring vector $\mathbf{w}_{h_j}(\mathbf{y}) \in \mathbb{R}^m$ can be constructed, ranking the m possible prediction candidates in descending order such that $w_1 \geq w_2 \geq \ldots \geq w_m$. In the context of link prediction using Knowledge Graph Embeddings (KGEs), this ranking is generated as previously described. For classification tasks, the ranking can be established by assigning a rank value of 1 to the predicted class and 0 to all others.
The resulting scoring vectors $\mathbf{w}$ can then be processed using three distinct methods from social choice theory:

Majority Voting: This method transforms the scoring vector $\mathbf{w}$ by assigning a value of 1 to w_1, while all other entries are set to 0.

Borda Voting: This method adjusts the scoring vector $\mathbf{w}$ such that,
$w_1 \to m-1, w_2 \to m-2, \ldots, w_m \to 0$, for m candidates.

Range Voting: This method normalizes the scoring vector, ensuring that $w_1 = 1$ and $w_m = -1$, by linear interpolation between the minimum and the maximum.

To mitigate predictive multiplicity, one of these three voting methods is selected, and the corresponding scoring vector is calculated for each of the l competing models for a given prediction $\mathbf{y}$. This results in l scoring vectors $\mathbf{w}$. These vectors are then combined to a final prediction, e.g. $\mathbf{y}_{\text{majority}}$, resulting in an ensemble learning-based approach. For instance, in the case of Majority Voting, the prediction most frequently ranked at position w_1 across all voting vectors is chosen as the final prediction.

	Link Prediction Models			Voting Methods aggregating h_i		
Rank	$h_0(q,e)$	$h_1(q,e)$	$h_2(q,e)$	Majority	Borda	Range
1	Earth (1.0)	Mars (109.5)	Mars (3.4)	Mars (3)	Mars (16)	Mars (3.0)
2	Jup. (1.0)	Jup. (77.7)	Jup. (3.1)	Earth (1)	Jup. (15)	Jup. (2.2)
3	Mars (1.0)	Earth (74.8)	Earth (2.8)	Jup (1)	Earth (14)	Earth (2.1)
4	Moon (0.4)	Hubble (50.4)	Moon (0.0)	Moon (0)	Moon (8)	Moon (-1.1)
5	Rover (0.3)	Moon (29.1)	Hubble (-1.9)	Hubble (0)	Hubble (6)	Hubble (-1.6)
6	Hubble (0.2)	Rover (25.3)	Rover (-1.9)	Sun (0)	Rover (4)	Rover (-1.9)
7	Sun (0.0)	Sun (20.6)	Sun (-2.9)	Rover (0)	Sun (0)	Sun (-3.0)

Table 1: Sorted values of $h_i(tr(q,e))$ with $q = \langle ?, \text{Orbits}, \text{Sun} \rangle$, where the columns are sorted in descending order according to the value $e \in E$. The horizontal lines after rank 4 indicate whether or not a entity is in the Top 4 prediction.

In the link prediction task from Section 3.4 the prediction of Majority Voting is Mars. The right side of Tab. 1 shows the result of Majority, Borda and Voting methods. The prediction via the Borda voting method is reached via

$$\mathbf{y}_{\text{Borda}} = \sum_{i=0}^{2} \tilde{\mathbf{w}}_{\text{major}}(h_i(q, E)) = \begin{pmatrix} 6 \\ 5 \\ 4 \\ 3 \\ 2 \\ 1 \\ 0 \end{pmatrix} + \begin{pmatrix} 4 \\ 5 \\ 6 \\ 2 \\ 1 \\ 3 \\ 0 \end{pmatrix} + \begin{pmatrix} 4 \\ 5 \\ 6 \\ 3 \\ 1 \\ 2 \\ 0 \end{pmatrix} = \begin{pmatrix} 14 \\ 15 \\ 16 \\ 8 \\ 4 \\ 6 \\ 0 \end{pmatrix} \tag{22}$$

where $\tilde{\mathbf{w}}$ sorts the weight vector $\mathbf{w}$ according to the order of entities of the baseline h_0 in the leftmost column in Tab. 1, i.e. Earth, Jupiter, Mars, Moon and so on. This enables aggregation via adding. The final prediction is achieved by sorting the aggregation.
As a result to mitigate PM Borda voting and Range voting prove successful. The simplification of Majority voting makes the decision for this small example where ranks below K have zero values ambiguous.

5 Analysis

5.1 Sources of Predictive Multiplicity

Predictive multiplicity arises from both data characteristics and model design. The structure and assumptions within a model, such as feature selection, hypothesis class misspecification, or the presence of latent groups, contribute to multiplicity [3]. Similarly, in Knowledge Graphs (KGs), entities or relations with fewer facts lead to greater uncertainty, resulting in multiple potential embeddings and conflicting predictions. More expressive models, which capture a broader range of embeddings, are also prone to higher multiplicity due to their flexibility in fitting the training data. [5]

5.2 Handling Predictive Multiplicity

To address predictive multiplicity, it is essential to measure and report it alongside other metrics like test error, ensuring context-sensitive solutions. This approach enhances transparency and decision-making by acknowledging multiplicity as an inherent aspect of model performance [9]. Voting and ensemble methods help mitigate multiplicity by reducing conflicting predictions, especially for less frequent or more uncertain entities and relations [5].

5.3 Overfitting: Implications for Predictive Multiplicity

In some cases, a classifier h (or M_θ) may perform well on the training data but poorly on the test set, indicating overfitting. This means the model has memorized the specific details of the training data rather than learning the general patterns, leading to arbitrary predictions for new, unseen data. Such models inherently exhibit high predictive multiplicity because their predictions lack consistency across different datasets. Therefore, it is crucial to eliminate overfitting models when examining predictive multiplicity to avoid confusing the effects of overfitting with those of true predictive multiplicity.

6 Conclusion

This paper provides a comprehensive examination of predictive multiplicity, highlighting its impact on classification and link prediction tasks. Leveraging clearly comprehensible, academic examples, we illustrate how predictive multiplicity manifests across models, potentially undermining reliability and fairness in critical applications. Exploring social choice theory as a mitigation strategy offers a path to more robust and interpretable outcomes. Our findings support adopting multiplicity-aware practices in model evaluation to better account for nuanced model behaviors. Further research is encouraged to refine mitigation techniques and assess predictive multiplicity's impact across machine learning domains, ensuring fair and reliable systems.

References

1. Black, E., Raghavan, M., Barocas, S.: Model Multiplicity: Opportunities, Concerns, and Solutions. In: Proceedings of the 2022 ACM Conference on Fairness, Accountability, and Transparency, pp. 850–863. ACM, New York, NY, USA (2022). https://doi.org/10.1145/3531146.3533149
2. Breiman, L.: Statistical Modeling: The Two Cultures (with comments and a rejoinder by the author). Statistical Science **16**(3), 199–231 (2001). https://doi.org/10.1214/ss/1009213726
3. Marx, C.T., du Pin Calmon, F., Ustun, B.: Predictive Multiplicity in Classification. arXiv preprint arXiv:1909.06677 (2020). https://arxiv.org/abs/1909.06677
4. National Institute of Justice: Predicting Recidivism: Continuing to Improve the Bureau of Prisons Risk Assessment Tool. https://nij.ojp.gov/topics/articles/predicting-recidivism-continuing-improve-bureau-prisons-risk-assessment-tool, last accessed 2024/11/29
5. Zhu, Y., Potyka, N., Nayyeri, M., Xiong, B., He, Y., Kharlamov, E., Staab, S.: Predictive Multiplicity of Knowledge Graph Embeddings in Link Prediction. arXiv preprint arXiv:2408.08226 (2024). https://arxiv.org/abs/2408.08226
6. A. Mathur and G. M. Foody, "Multiclass and Binary SVM Classification: Implications for Training and Classification Users," in IEEE Geoscience and Remote Sensing Letters, vol. 5, no. 2, pp. 241-245, April 2008, doi: 10.1109/LGRS.2008.915597.
7. Ge, X., Wang, Y.-C., Wang, B., Kuo, C.-C. J.: Knowledge Graph Embedding: An Overview. arXiv preprint arXiv:2309.12501 (2023). https://arxiv.org/abs/2309.12501
8. Bordes, A., Usunier, N., et al.: Translating Embeddings for Modeling Multi-relational Data. In: Burges, C.J., Bottou, L., Welling, M., Ghahramani, Z., Weinberger, K.Q. (eds.) Advances in Neural Information Processing Systems, vol. 26. Curran Associates, Inc. (2013).

9. Watson-Daniels, J., Parkes, D. C., Ustun, B.: Predictive Multiplicity in Probabilistic Classification. arXiv preprint arXiv:2206.01131 (2023). https://arxiv.org/abs/2206.01131