Bayesian Approach to Uncertainty Visualization of Heterogeneous Behaviors in Modeling Networked Anagram Games

Xueying Liu[1], Zhihao Hu[1], Xinwei Deng[1], and Chris J. Kuhlman[2]

[1] Virginia Tech, Blacksburg VA 24061, USA,
xliu96@vt.edu, huzhihao@vt.edu, xdeng@vt.edu,
[2] University of Virginia, Charlottesville VA 22904, USA,
hugo3751@gmail.com

Abstract. Heterogeneous player behaviors are commonly observed in games. It is important to quantify and visualize these heterogeneities in order to understand collective behaviors. Our work focuses on developing a Bayesian approach for uncertainty visualization in a model of networked anagram games. In these games, team members collectively form as many words as possible by sharing letters with their neighbors in a network. Heterogeneous player behaviors include great differences in numbers of words formed and the amount of cooperation among networked neighbors. Our Bayesian approach provides meaningful insights for inferring worst, average, and best player performance within behavioral clusters, overcoming previous model shortcomings. These inferences are integrated into a simulation framework to understand the implications of model uncertainty and players' heterogeneous behaviors.

Keywords: agent-based simulation, interpretable inference, models of heterogeneous behaviors, networked data, uncertainty visualization

1 Introduction

1.1 Background and Motivation

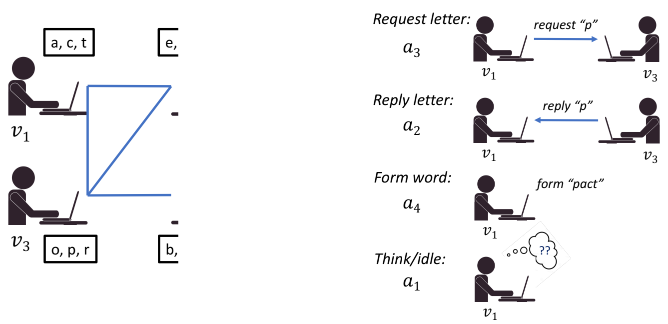
There are many variants of anagram games. Most involve either the unscrambling of letters to form a single unique word or finding as many words as possible from a collection of letters. Anagram games involving single individuals have been studied for over 50 years. As an early example from the 1960s, anagram games were used as priming activities to study anxiety [16], i.e., anagram games are played in a way to induce player anxiety. Also dating from the 1960s, these games have been studied in their own right, e.g., to assess the effects of letter rearrangement and word frequency on player performance [6].

Evaluation of group anagram games, where players cooperate to form words, is a much more recent phenomenon. A ground-breaking work in [2] used in-person group anagram games to prime people to form collective identity. The work in [1] performed *online* group anagram games by imposing a *network* on game players to control their interactions. Our focus is to model the games in [1], which we now overview.

The experimental networked group anagram game (NGrAG) setup is shown in Figure 1a. Remote human subjects play the game through web browsers. Each player is provided three initial letters and over a 5-minute game duration, players try to form as many words as possible *as a team*. Players split evenly the total earnings from the game, which is based on the number of words the team forms. Players cooperate by sharing their available letters with their distance-1 neighbors. When a player v_l shares a letter, she retains a copy of the letter; this is to motivate (mutual) assistance among players. Also, once a letter is acquired, it can be used any number of times in one word and can be used in any number

player loses a letter).

e any of the actions in



(a) online game configuration (b) 4 possible player actions

Fig. 1: (a) Illustrative networked group anagram game (NGrAG) with four remote players (v_1 through v_4) and four communication channels (in blue). Players participate through their web browsers. A player's initially assigned letters are in boxes. (b) Four actions that may be taken by any player, at any time during the 5-minute NGrAG. Actions can be repeated by a player any number of times. The action vector a is $a = (a_1, a_2, a_3, a_4)$, with a_i given in the graphic.

The network has at least three roles in NGrAGs, and these are intertwined with game player behavior (i.e., action) models. First, the network determines the number of neighbors (i.e., degree) of an (ego) player. Section 2.1 states how ego game player data are partitioned by degree in developing behavior models. Second, the letters assigned to those neighbors, along with those of the ego player, determine the words an ego player can form. Third, the behaviors of the neighbor nodes are influenced by their degrees—as for the ego player, in the first point—and hence these neighbors' interactions with the ego (e.g., requesting letters) are dependent on their local network structure.

1.2 Novelty and Contributions

Modeling of NGrAGs is an interesting and challenging task. A Bayesian modeling approach is described in [13]. Here, we focus on characterizing variability in player performance through Bayesian uncertainty visualization. Our contributions follow.

First, because of using posterior samples without asymptomatic distributions of model parameters (as in [9]), the proposed Bayesian uncertainty visualization can greatly alleviate the data scarcity issue in model estimation. Consequently, the obtained posterior samples of model parameters avoid extreme values that cause some transition probabilities π_{ij} to be 0 or 1. See Sections 2 and 4 and Figure 2.

Second, the proposed Bayesian uncertainty visualization is a first work to appropriately visualize the uncertainties in data and models for NGrAGs. Different from previous work [9], the proposed method emphasizes the visualization of uncertainty in a comprehensive manner using a two-dimensional bubble plot described in Section 3. Such a plot can reflect uncertainties of a player's activeness (i.e., non-idle actions). Moreover, the location, width, and height of bubbles represent the mean and standard deviations of the probabilities inferred from the posterior samples of model parameters. Plots of results from our experiments and analyses are in Section 4.

Third, uncertainty visualization enhances agent-based modeling and simulation (ABMS) of these games. We can naturally identify, interpret, and model worst, average, and best categories of player performance, even within one cluster of player behavior. We embed these interpretable inferences within a simulation platform. We demonstrate these effects by simulating NGrAGs that go beyond the conditions for which experiments were conducted (Section 5). The network of each simulated game is fixed, consistent with the experiments being modeled.

Our last contribution is broader in scope. The proposed Bayesian uncertainty visualization greatly enhances the explainability of uncertainty quantification for NGrAGs. For a complex system such as a NGrAG, in contrast to "one-shot" games where a game player only makes one binary yes/no decision in a game, quantifying uncertainty for model and data needs to be properly visualized in order to gain meaningful insights. It is precisely this need that motivated this work, which is an outgrowth of the work in [9]. The proposed method can be a good exemplar to achieve such an objective, and can be applied for visualizing uncertainty in other networked applications. Specific works are provided in Related Work Section 1.3 below.

1.3 Related Work

Modeling of network games and data. There are multiple works [9] [13] related to the proposed method. In [13], a Bayesian model of human behavior in anagram games is presented. However, that work does not address methods to identify worst, average, and best behaviors within a Bayesian context. A process of identifying best, average, and worst behaviors within behavioral clusters is presented in [9]. However, that anagram model for determining a player's next action in a game is based on asymptotic normal distributions for the primary behavioral matrix $\boldsymbol{B}$ (presented below), rather than on posterior distributions, as we do here. Most importantly, neither of those works address uncertainty visualization, as in this work. Other games incorporate multiple player actions over time, e.g., [11, 17]. These games, like ours, use fixed networks. Other types of network models, for other phenomena, use evolving networks, e.g., [5].

Bayesian visualization and uncertainty visualization. Visualization is a vital tool for data analysis to describe uncertainties in data [3, 4]. The effective visualization of uncertainty is commonly recognized as a challenging task [10, 14]. Potter et al. [15] presented a summary of the state-of-the-art techniques in uncertainty visualization, including comparison techniques, attribute modification, and image discontinuity. Gabry et al. [7] illustrated the role of visualization in exploratory data analysis in the context of a Bayesian workflow. House et al. [8] developed Bayesian visual analytics (BaVA) to justify Bayesian sequential update of parameters. Our work aims to visualize uncertainty in a Bayesian framework to effectively and accurately identify the uncertainty in the data and heterogeneous behaviors of players.

2 State Transition Model and Extension

2.1 State Transition Model

Agent-based models (ABMs) for the NGrAG represent the game as a discrete-time stochastic process. That is, at each time step, a player can transition to one of the four states (actions a_1, a_2, a_3, and a_4); see Figure 1b. In our previous work [13], a Bayesian clustering-based UQ framework is developed as follows. Based on statistical analysis of the game data, we first partitioned the players into two groups: those with less than three neighbors (group $g = 1$) and those with three or more neighbors (group $g = 2$). Then we defined two variables x_e (for engagement) and x_w (for forming words), where x_e is the sum of the number of requests and the number of replies that a player sends and x_w is the number of words a player forms in a game. Based on these two standardized variables, we applied the Dirichlet process (DP)-based Bayesian clustering approach [12] with a specific penalty parameter λ ($\lambda = 2.5$) such that when a point is farther than λ away from every existing cluster center, a new cluster will be formed with this point in it. In this way, we further partitioned the players in the same group into four clusters where those within the same cluster have similar activity levels in the game. For data in each cluster, player behaviors in a game are modeled using the multinomial logistic regression with four predictors shown in Table 1:

$$\pi_{ij} = \exp(\mathbf{z}^T \boldsymbol{\beta}_j^{(i)}) / \sum_{m=1}^{l} \exp(\mathbf{z}^T \boldsymbol{\beta}_m^{(i)}),\ j = 1, \ldots, l, \tag{1}$$

where

- $l = 4$ since we consider four actions a_1, a_2, a_3, and a_4.
- π_{ij} is the probability of the player, who took action a_i at time t, taking action a_j at time $t + 1$.
- $\mathbf{z} = (1, Z_B(t), Z_L(t), Z_W(t), Z_C(t))^T$ is the predictor vector; Table 1.

– $\boldsymbol{\beta}_j^{(i)} = (\beta_{j1}^{(i)}, \dots, \beta_{j5}^{(i)})^T$ is the regression coefficient parameter vector.

Variable	Description
$Z_B(t)$	Size of buffer of letter requests that player v has yet to reply to at time t.
$Z_L(t)$	Number of letters that v has available to use at t to form words.
$Z_W(t)$	Number of valid words that v has formed up to t.
$Z_C(t)$	Number of consecutive time steps that v has taken the same action.

Table 1: The four temporal variables of players in the NGrAG and model.

For a given action a_i at time t, the parameters can be expressed as a matrix $\boldsymbol{B}^{(i)} = (\boldsymbol{\beta}_1^{(i)}, \dots, \boldsymbol{\beta}_l^{(i)})^T_{l\times(l+1)}$ for $i = 1, \dots, 4$. Thus, from the game network structure (which determines a node's degree and hence its group g) and the performance cluster c determined for a node/player ($c = 1$, 2, 3, or 4), a particular model based on Equation (1), with parameter matrix $\boldsymbol{B}^{(i)}$, is assigned to a game player to predict the probability of next actions.

2.2 Motivation for Model Extension

With the Bayesian approach, we can obtain the posterior distribution for the parameter matrix $\boldsymbol{B}^{(i)}$. One then can quantify the uncertainty of parameters by conducting posterior inference. Markov chain Monte Carlo (MCMC) methods are commonly used to obtain samples from the posterior distribution. Posterior inference can then be conducted empirically. In order to quantify the heterogeneous behavior of players *within* a cluster, our strategy is to identify the parameter matrices that generate the most active behavior, the least active behavior, and the average behavior in terms of probability of being non-idle. Integrating these different levels of performance into the simulation of NGrAG will better capture the heterogeneity among players because we can assign to players these different behaviors. These considerations lead to the new uncertainty visualization method in Section 3.

3 Bayesian Uncertainty Visualization Method

This section details the proposed Bayesian uncertainty visualization method. The goals are to visualize the uncertainty within and between clusters and to identify the heterogeneous (i.e., worst, average, and best) behaviors of players within each cluster.

For each observation in the training data, one can obtain the corresponding predictor vector $\mathbf{z}$. To directly identify the activity level, we transform the parameter matrix $\boldsymbol{B}^{(i)}$ to a probability vector. In each cluster, players with the same initial action a_i share the same parameter matrix $\boldsymbol{B}^{(i)}$. Thus for these players, without loss of generality, we omit i in $\boldsymbol{B}^{(i)}$ and π_{ij} to get a parameter matrix $\boldsymbol{B}$ and a probability vector $\boldsymbol{\pi} = (\pi_1, \dots, \pi_4)$ containing the probabilities of the next action using Equation (1). Therefore, for a $\boldsymbol{B}$ matrix, a training data set of n observations that have same initial action can generate n probability vectors. The mean of these probability vectors and the corresponding standard error can be obtained. Given a sequence of $\boldsymbol{B}$ matrices, we can compute a sequence of mean probability vectors and their standard errors. To visualize the uncertainty among these mean probability vectors, we create a bubble plot where the center of each bubble represents the mean probability vector for a parameter matrix, with the width to be $2 \times SE(\bar{\pi}_4^r)$ and the height to be $2 \times SE(\bar{\pi}_1^r)$. Using such a plot, it is easy to quantify the activity levels within a cluster and identify the worst, average, and best behaviors. The probability of forming words (π_4) in the probability vector represents the players' ability to form words and the probability of not being idle ($1 - \pi_1$) indicates the players' level of activity in the game—a small to-idle probability π_1 suggests a high activity level.

We summarize the proposed method of uncertainty quantification within a cluster as follows. First, we use Metropolis-Hasting (M-H) algorithm to get R random samples of $\boldsymbol{B}_r$ ($r = 1, \ldots, R$) from the posterior distribution after a burn-in period (taken to be 1000). Second, for each $\boldsymbol{B}_r$, we apply the size n training data to Equation (1) to produce n probability vectors $\hat{\boldsymbol{\pi}}^{r,l} = (\hat{\pi}_1^{r,l}, \hat{\pi}_2^{r,l}, \hat{\pi}_3^{r,l}, \hat{\pi}_4^{r,l})$, $l = 1, \ldots, n$. Then the mean probability vector and its standard error are calculated:

$$\bar{\boldsymbol{\pi}}^r = \frac{1}{n} \sum_{l=1}^{n} \hat{\boldsymbol{\pi}}^{r,l} = (\bar{\pi}_1^r, \bar{\pi}_2^r, \bar{\pi}_3^r, \bar{\pi}_4^r)^T,$$

$$SE(\bar{\boldsymbol{\pi}}^r) = \frac{1}{n} \sqrt{\sum_{l=1}^{n} (\hat{\boldsymbol{\pi}}^{r,l} - \bar{\boldsymbol{\pi}}^r)^2} = (SE(\bar{\pi}_1^r), SE(\bar{\pi}_2^r), SE(\bar{\pi}_3^r), SE(\bar{\pi}_4^r))^T.$$

Third, one can draw a bubble plot of $1 - \bar{\pi}_1$ against $\bar{\pi}_4$, with one bubble for each $\boldsymbol{B}_r$, $r = 1, \ldots, R$. Each bubble is an ellipse centered at the mean probability $(\bar{\pi}_4^r, 1 - \bar{\pi}_1^r)$ with width $2 \times SE(\bar{\pi}_4^r)$ and height $2 \times SE(\bar{\pi}_1^r)$. Fourth, note that a low mean to-idle probability ($\bar{\pi}_1$) suggests a high engagement level, and a player with a high mean to-idle probability is less active. Accordingly, we select the $\boldsymbol{B}_r$ matrix with the maximum $\bar{\pi}_1^r$ as the matrix of the worst behavior, and the one with the minimum $\bar{\pi}_1^r$ as the matrix of the best behavior. The $\boldsymbol{B}_r$ matrix of the average behavior is one that produces the mean of $\bar{\pi}_1^r$, $r = 1, \ldots, R$.

A key advantage of this proposed method is that we can visually analyze the uncertainty among data. In the bubble plot, it is easy to find the best and the worst behavior and view the heterogeneous behaviors within each cluster. Moreover, the size of the bubble can help us visually detect the variability among the observations. One can also quantitatively compare the activity ranges of clusters with players that have the same number of neighbors to further discover the differences between clusters within the same group ($g = 1$ or 2). Note that this visualized uncertainty quantification was not contained in the previous work [9].

Another advantage is that our Bayesian method alleviates the extreme value problem caused by data scarcity in the previous model [9]. When the size of the training data in each category is unbalanced (e.g., 556 observations have final state idle while only 4 observations have final state reply (a_2) and request (a_3) in group $g = 1$ cluster $c = 2$ with initial state a_3), the asymptotic normal distribution of $\boldsymbol{B}$ would have a very large variance. Thus, the estimated parameter in $\boldsymbol{B}$

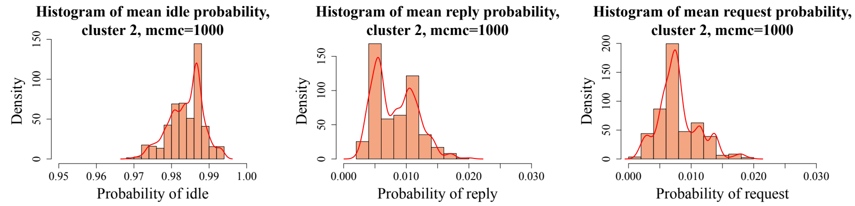


Fig. 2: The three histograms are for group 1, cluster 2, where the initial state is request (a_3). M-H algorithm is applied to draw 1000 $\boldsymbol{B}$ matrix samples after 1000 burn-in. The first histogram is for the probability of transitioning to idle (π_1), the second one is for the probability of transitioning to reply (π_2), and the third one is for the probability of request (π_3). The probability of forming words (π_4) is 0 because there is no forming words action a_4 in the training data. These data demonstrate that the extreme value problem is largely alleviated.

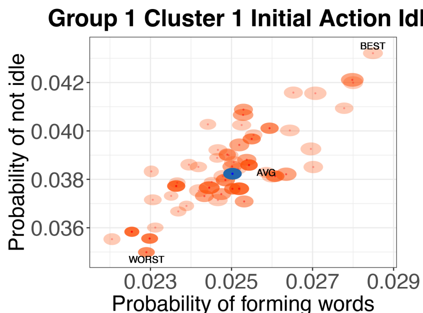
can be unexpectedly large and cause an extreme value in the probability vector $\boldsymbol{\pi}$ and an infinite loop in state transitions in the ABM. However, the memorylessness property of MCMC can avoid this problem since every sample is only generated based on the previous one. For this reason, the Bayesian approach avoids the extreme scenarios of players' actions.

4 Visualization of Heterogeneous Behaviors

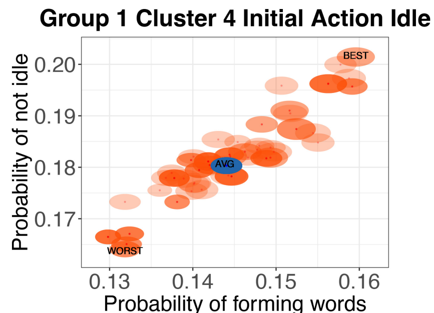
This section investigates uncertainties within clusters, heterogeneous behaviors, and differences in activity levels between clusters, using the game data and the models of Sections 2 and 3. Under the Bayesian setting, for each initial state in a cluster, 1000 samples of $\boldsymbol{B}$ matrices are drawn using the M-H algorithm after 1000 burn-in. Figure 2 reports the histograms of probabilities for the aforementioned group 1, cluster 2 with initial state being request (a_3). It is seen that the Bayesian uncertainty quantification methods can alleviate extreme value problems (by producing probabilities away from 0 and 1) caused by data scarcity.

The bubble plots for group 1, cluster 1 and cluster 4, with the initial state being idle (a_1) are presented in Figures 3a and 3b, respectively. Clearly, there is uncertainty within the clusters. Moreover, the size of the bubble reflects the variability in the observations and the color reflects the replications. The darker the bubble, the more samples have this transition probability. It is seen in each plot that samples are more gathered at the maximize-a-posterior (MAP) estimation (the blue bubble) and the standard error of to-word (transition) probability is larger than that of to-idle probability in most cases.

Figure 4 contains bubble plots of mean probability for initial state idle (a_1) in the top row, and for initial state reply (a_2) in the bottom row. When the initial state is idle, Figures 4a and 4b show that four clusters are well separated and the activity level is ascending, supporting the rationality of clustering players by behavior. It is also seen that group 2 is more active than group 1 with a larger probability of forming words and being non-idle, on a per cluster basis. This corresponds to our assumption that players with more neighbors will be



(a)



(b)

Fig. 3: (a) Bubble plot of group 1, cluster 1, where initial state is idle (a_1). (b) Bubble plot of group 1, cluster 4, where initial state is idle (a_1). Each bubble is a ellipse centered at the mean probability $(\bar{\pi}_4^r, 1-\bar{\pi}_1^r)$ with width $2 \times SE(\bar{\pi}_4^r)$ and height $2 \times SE(\bar{\pi}_1^r)$, where $SE(\bar{\pi}_4^r)$ and $SE(\bar{\pi}_1^r)$ are standard errors of mean to-word probability and mean to-idle probability, respectively. The blue bubbles are the MAP results. The worst, average, and best performance bubbles are noted.

more active in the NGrAG. When the initial state is reply (second row of plots), there are no data points of forming words in the training data of group 1. Thus, the mean probabilities of forming words and the corresponding standard errors will be zero. Consequently, we compare the activity level only based on the probability of being non-idle as shown in Figure 4c.

5 Agent-Based Simulations of Networked Anagram Games and Results

In this section, we build ABMs and run them in a software framework to simulate the NGrAG. For particular input conditions and models, we provide results for individual players (also referred to as nodes or agents) and for aggregated totals over all players.

5.1 Simulation Process

ABMs are designed and constructed from the models of Sections 2 and 3. Inputs to simulations are as follows. The network of Figure 5 represents the possible interactions among the seven game players. It contains players in groups $g = 1$ and 2. Each player is provided four letters, and the letters are purposely specified to enable players to form words, e.g., one player v_2 is given letters $\{i, l, m, n\}$ and neighboring players are given complementary letters, e.g., v_3 is assigned letters $\{o, p, r, s\}$. Owing to space considerations, we examine two clusters: cluster $c = 3$ for $g = 1$ and cluster $c = 2$ for $g = 2$. With these clusters, we then execute the worst, average, and best behavior models that are produced and evaluated in Sections 3 and 4. Note that our results illustrate differences among worst, average, and best models, but the results shown are not the largest differences

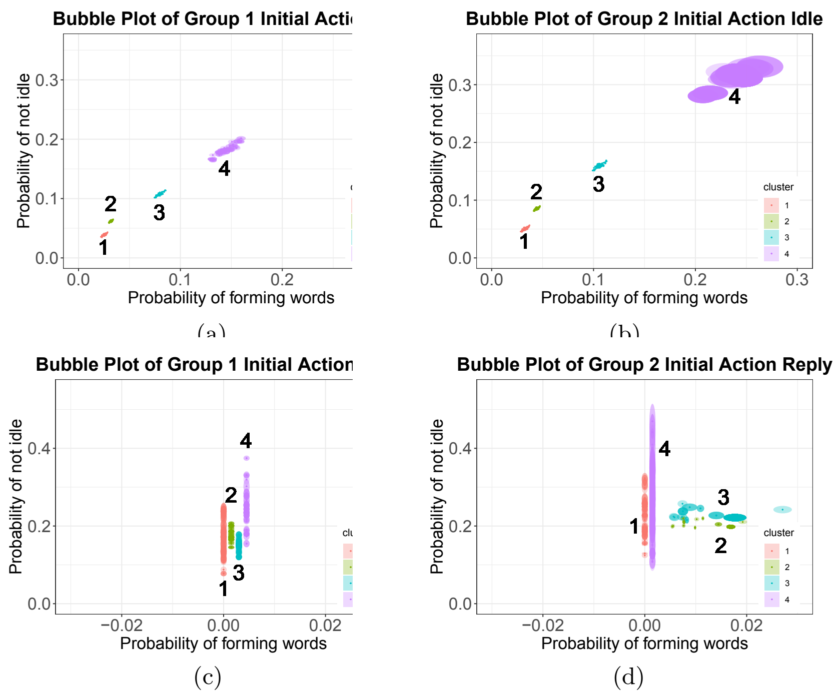


Fig. 4: (a) Bubble plot for group 1 where initial action is idle (a_1). The bubbles for cluster 1 and cluster 4 are shown in Fig. 3a and Fig. 3b, respectively. (b) Bubble plot for group 2 where initial action is idle (a_1). (c) Bubble plot for group 1 where initial action is reply (a_2). Note that the probabilities of forming words are 0 for all clusters. We assign different values for bubbles in different clusters to avoid overlapping. (d) Bubble plot for group 2 where initial action is reply (a_2). Note that the probabilities of forming words are 0 for cluster 1 and cluster 4. We assign a different value for bubbles in cluster 4 to avoid overlapping.

that exist across all $[g, c]$ pairs. This is to emphasize that the differences that we observe from the simulations are pervasive across all model parameters; an expanded version will address the full range of results.

One simulation is comprised of 100 iterations or runs. Each iteration is a complete simulation of one NGrAG, from time $t = 0$ to $t = 300$ seconds where players request letters from neighbors, reply to neighbor letter requests, and form words, as in the experiments. Our time step is one second, justified by the fact that players do not take successive actions among request letter, reply to letter request, and form word within one second in the online experiments. Indeed, actions at time steps are mostly idle or thinking. The difference among iterations within a simulation is the stochasticity of the models in producing probabilities of actions π_{ij} of players at each t. Consequently, when we refer to "average" results below, we mean time point-wise averages across the 100 iterations, unless specified otherwise. Finally, we use the "worst," "average,"

and “best” se the same
model, so t ited by the
best model st model.

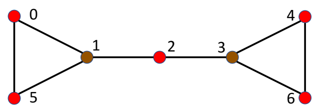


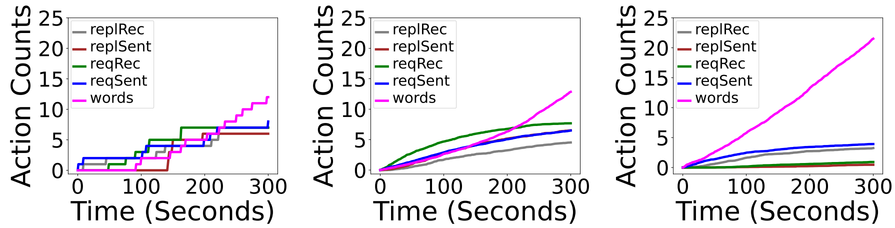
Fig. 5: Seven node (agent) game network on which simulations are run. Red (resp., brown) nodes are low (resp., high) degree nodes of degree $d = 2$ (resp., $d = 3$). Therefore, red (resp., brown) nodes are in group $g = 1$ (resp., $g = 2$).

5.2 Visualization of Simulation Results

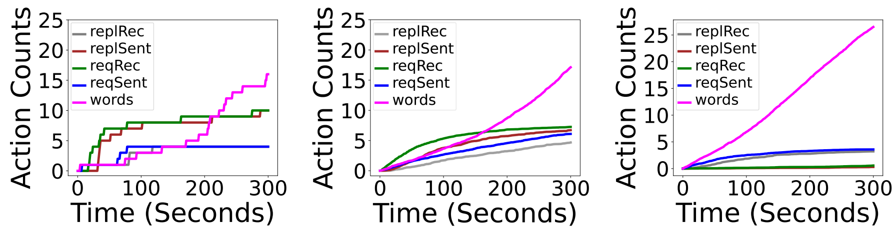
Figure 6 provides results for a degree $d = 3$ player (agent 3) and a $d = 2$ player (agent 5) from the game setup in Figure 5. The $[g, c]$ values are given in the caption of Figure 6. Data in the first row of plots were generated with the worst behavior models for the respective $[g, c]$ pairs. In Figure 6a, the time histories of actions are given for one of the 100 iterations: number of replies received (rerplRec), of replies sent (replSent), of requests received (reqRec), of requests sent (reqSent), and of words formed (words). The stair-stepped nature of the curves is due to the fact that these curves are from one iteration and so the plotted actions are discrete. In Figure 6b, the curves correspond to the same action histories, but are smoother because they represent the time point-wise average of all 100 iterations. These first two plots are for agent 3. Figure 6c depicts corresponding average data for player 5. Note that a player with fewer neighbors (player 5) forms more words than a player with a greater number of neighbors (player 3). This is because player 5’s behavior is from cluster 3 of group 1, while player 3’s behavior is from cluster 2 of group 2; e.g., see Figures 4a and 4b and the x-axis values for the two clusters. This again demonstrates the efficacy of identifying heterogeneous behaviors of players.

Figures 6d through 6f provide the corresponding plots to those in the first row, but now the results are for the best model. In Figure 6e for player 3, the number of words is greater than that for the worst model, although the numbers of sharing actions are about the same. This same comparison holds for player 5 in Figure 6f versus Figure 6c.

Figure 7 contains aggregate data over all seven game players. Time histories of the total number of words formed for the worst, average, and best models are given in Figure 7a; the numbers of words increase in this order of the models. Figure 7b provides similar data for the sharing actions of requests and replies, but now only for the worst (dashed curves) and best (solid curves) models. Now, the worst and best model results overlap, with no clear-cut better behavior. This is partially a consequence of the fact that numbers of requests and replies are bounded by the number of a player’s neighbors and the number of letters per player. Figure 7c shows the time point-wise *average* probability over all players of taking each action. These probabilities reflect the action counts in the previous two plots.

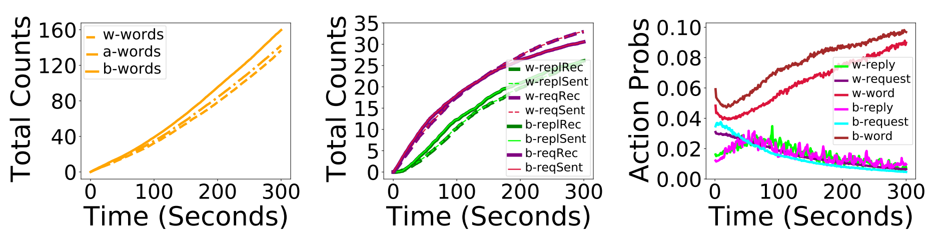


(a) worst: agent 3, one iter (b) worst: agent 3, ave (c) worst: agent 5, ave



(d) best: agent 3, one iter (e) best: agent 3, ave (f) best: agent 5, ave

Fig. 6: Results of anagram simulations with seven players. Data in plots (a) through (c) are for the *worst* behavior model. The curves are game time histories of counts of actions over the 300 second game. (a) action histories for agent 3 in one iteration, (b) average action histories for agent 3, and (c) average action histories for agent 5. Data in plots (d) through (f) are the respective plots for the *best* behavior model.



(a) sum of words formed (b) sum of requests & replies (c) action probabilities

Fig. 7: Aggregate simulation results across all seven nodes (agents) in a NGrAG.(a) Sum of words formed by all players in time for worst (w-words), average (a-words), and best (b-words) behavior models. (b) Sum of requests and replies across all players. Dashed curves correspond to worst behavior model (prefix "w-" in legend) and solid curves correspond to best behavior model (prefix "b-" in legend). (c) Average action probabilities across all seven players in a game, in time, for replying to letter requests (reply), requesting letters (request), and forming words (word).

6 Summary

This work presents a Bayesian uncertainty visualization method of complicated multi-player game data. Our step-by-step procedures have been applied to a networked group anagram game, where players cooperate to share letters and form words. These visualizations can effectively assist in assessing model uncertainties, and in improving the interpretable inference of player behaviors. Software modules of these models are used to simulate the game for conditions beyond the experiments.

References

1. Cedeno, V., Hu, Z., et al.: Networked experiments and modeling for producing collective identity in a group of human subjects using an iterative abduction framework. Soc. Netw. Anal. Min. (SNAM) p. 43 (2020)
2. Charness, G., Cobo-Reyes, R., et al.: Identities, selection, and contributions in a public-goods game. Games and Economic Behavior (2014)
3. Chen, C.h., Härdle, W.K., Unwin, A.: Handbook of data visualization. Springer Science & Business Media (2007)
4. Cook, D., Lee, E.K., Majumder, M.: Data visualization and statistical graphics in big data analysis. Annual Review of Statistics and Its Application 3, 133–159 (2016)
5. Dankulov, M.M., Melnik, R., Tadić, B.: The dynamics of meaningful social interactions and the emergence of collective knowledge. Scientific Reports 5, 12197–1–12197–10 (2015)
6. Dominowski, R.L.: Anagram solving as a function of letter moves. Journal of Verbal Learning and Verbal Behavior 5, 107–111 (1966)
7. Gabry, J., Simpson, D., Vehtari, A., Betancourt, M., Gelman, A.: Visualization in Bayesian workflow. Journal of the Royal Statistical Society: Series A (Statistics in Society) 182(2), 389–402 (2019)
8. House, L., Leman, S., Han, C.: Bayesian visual analytics: Bava. Statistical Analysis and Data Mining: The ASA Data Science Journal 8(1), 1–13 (2015)
9. Hu, Z., Deng, X., Kuhlman, C.J.: Versatile uncertainty quantification of contrastive behaviors for modeling networked anagram games. In: Complex Networks (2021)
10. Johnson, C.R., Sanderson, A.R.: A next step: Visualizing errors and uncertainty. IEEE Computer Graphics and Applications 23(5), 6–10 (2003)
11. Kearns, M., Suri, S., Montfort, N.: An experimental study of the coloring problem on human subject networks. Science 313(5788), 824–827 (2006)
12. Kulis, B., Jordan, M.: Revisiting k-means: New algorithms via Bayesian nonparametrics. In: ICML (2012)
13. Liu, X., Hu, Z., Deng, X., Kuhlman, C.J.: A Bayesian uncertainty quantification approach for agent-based modeling of networked anagram games. In: WSC (2022)
14. Pang, A.T., Wittenbrink, C.M., Lodha, S.K., et al.: Approaches to uncertainty visualization. The Visual Computer 13(8), 370–390 (1997)
15. Potter, K., Rosen, P., Johnson, C.R.: From quantification to visualization: A taxonomy of uncertainty visualization approaches. In: IFIP Working Conference on Uncertainty Quantification. pp. 226–249. Springer (2011)
16. Russell, D.G., Sarason, I.G.: Test anxiety, sex, and experimental conditions in relation to anagram solution. J. Personality and Social Psychology pp. 493–496 (1965)
17. Unakafov, A.M., Schultze, T., et al.: Emergence and suppression of cooperation by action visibility in transparent games. PLoS Comput Biol 16(1), e1007588 (2020)