Evaluation of Measurement Techniques for the Validation of Agent-Based Simulations Against Streaming Data

Timothy W. Schoenharl and Greg Madey

Department of Computer Science and Engineering
University of Notre Dame
Notre Dame, IN 46556

Abstract. This paper presents a study evaluating the applicability of several different measures to the validation of Agent-Based Modeling simulations against streaming data. We evaluate the various measurements and validation techniques using pedestrian movement simulations used in the WIPER system. The Wireless Integrated Phone-based Emergency Response (WIPER) system is a Dynamic Data-Driven Application System (DDDAS) that uses a stream of cellular network activity to detect, classify and predict crisis events. The WIPER simulation is essential to classification and prediction tasks, as the simulations model human activity, both in movement and cell phone activity, in an attempt to better understand crisis events.[1]

1 Introduction

Crisis events occur without warning on short time scales relative to normal human activities. Simulations designed to model human behavior under crisis scenarios faces several challenges when attempting to model behavior in real time. The DDDAS approach seeks to address that challenge by incorporating real time sensor data into running simulations[1][2]. Dynamic, Data-Driven Application Systems is an approach to developing systems incorporating sensors and simulations where the simulations receive streaming data from the sensors and the sensors receive control information from the simulations.

The WIPER system uses streaming cell phone activity data to detect, track and predict crisis events [3]. The Agent-Based Modeling simulations in the WIPER system are intended to model the movement and cell phone activity of pedestrians. These simulations model crisis events and are intended to be validated in a online fashion against streaming data.

In the WIPER system, ensembles of simulations are created, with each simulation parameterized with a particular crisis scenario and initialized from the streaming data. When the all of the simulations in the ensemble have finished

[1] The research on WIPER is supported by an NSF Grant, CISE/CNS-DDDAS, Award #0540348.

M. Bubak et al. (Eds.): ICCS 2008, Part III, LNCS 5103, pp. 6–15, 2008.

running, the results are validated against streaming data from the cell phone network.

Thus the validation technique must provide a method of discriminating between various simulations. In the context of the WIPER project, this means determining which crisis model is the best fit for the phenomenon detected in the streaming data. In this paper we will demonstrate a validation technique and evaluate the technique with several measures.

2 Background

Validation is described as the process of determining whether a given model is an appropriate choice for the phenomenon being modeled. In the model development cycle, validation is normally considered in the context of simulation creation and development, and is done nearly exclusively in an offline fashion. However, the process of selecting an appropriate model for a given phenomenon is precisely what is needed in the dynamic context of the WIPER system.

There exists a large body of work on the topic of simulation validation. A survey of techniques and approaches to offline validation of discrete event simulations can be found in Balci [4]. This work is an essential reference for validation, but many of the techniques are suited to offline validation only, as the interpretation requires human judgement.

This section is divided into three subsections related to the provenance of the techniques we intend to evaluate. The first section deals with canonical offline validation techniques from simulation, the second section presents distance measures and the third section presents work that has been done with online simulations.

2.1 Offline Simulation Validation

Balci presents a thorough evaluation of techniques for validation of models in the context of model and simulation development [4]. The intent for these techniques was to aid in the validation and verification of simulations prior to deployment. Some techniques mentioned by Balci that are useful to our current discussion are predictive validation (also called input-output correlation) and blackbox testing.

Kennedy and Xiang describe the application of several techniques to the validation of Agent-Based Models [5,6]. The authors separate techniques into two categories: subjective, which require human interpretation, and objective, for which success criteria can be determined *a priori*. We focus on objective techniques, as the requirements of a DDDAS system make it impossible to place human decision makers "in the loop".

2.2 Online Simulations

Researchers in the area of discrete event simulations recognize the challenges posed to updating simulations online from streaming data [7]. The need for

human interpretation is a serious limitation of traditional validation approaches and limits their usefulness in the context of online validation. Simulation researchers have defined a need for online validation but recognize the challenges to the approach. Davis claims that online validation may be unobtainable due to the difficulty in implementing changes to a model in an online scenario. We present a limited solution to this problem by offering multiple models simultaneously and using validation to select among the best, rather than using online validation to drive a search through model space.

It is important to distinguish between the model, the conceptual understanding of factors driving the phenomenon, and the parameters used to initialize the model. Optimization via simulation is a technique that is similar to canonical optimization and seeks to make optimal choices on selecting input parameters while keeping the underlying model the same and uses a simulation in place of the objective function. These techniques are usually grouped by whether they are appropriate for discrete or continuous input spaces [8]. For simulations with continuous input parameters, the author suggests the use of gradient-based methods. For simulations with discrete input parameters, the author presents approaches using random search on the input space.

3 Measures

We evaluate the following measures in the context of ranking simulations:

- Euclidean distance
- Manhattan distance
- Chebyshev distance
- Canberra distance

The distance measures are used to evaluate the output of the WIPER simulations in the context of agent movement. Agents move on a GIS space and agent locations are generalized to the cell tower that they communicate with. The space is tiled with Voronoi cells [9] that represent the coverage area of each cell tower. Empirical data from the cellular service provider aggregates user locations to the cell tower and the WIPER simulations do the same. Thus we can evaluate the distance measures using a well-defined vector of cell towers where the value for each position in the vector is the number of agents at that tower at each time step.

3.1 Distance Measures

Euclidean distance is the well-known distance metric from Euclidean geometry. The distance measure can be generalized to n dimensions from the common 2 dimensional case. The formula for Euclidean distance in n dimensions is given in Equation 1

$$d(\overline{p}, \overline{q}) = \sqrt{\sum_{i=1}^{n} (p_i - q_i)^2} \tag{1}$$

where

$$\overline{p} = (p_1, p_2, \cdots p_n) \tag{2}$$

$$\overline{q} = (q_1, q_2, \cdots q_n) \tag{3}$$

Manhattan distance, also known as the taxicab metric, is another metric for measuring distance, similar to Euclidean distance. The difference is that Manhattan distance is computed by summing the absolute value of the difference of the individual terms, unlike Euclidean distance which squares the difference, sums over all the differences and takes the square root. From a computational perspective Manhattan distance is significantly less costly to calculate than Euclidean distance, as it does not require taking a square root. The formula for Manhattan distance in n dimensions is given in Equation 4.

$$d(\overline{p}, \overline{q}) = \sum_{i=1}^{n} |p_i - q_i| \tag{4}$$

Chebyshev distance, also called the L_∞ metric, is a distance metric related to Euclidean and Manhattan distances [10]. The formula for the Chebyshev distance is given in Equation 5. The Chebyshev distance returns the maximum distance between elements in the position vectors. For this reason the metric seems appropriate to try on the WIPER simulations, as certain models may produce an output vector with one cell having a large variation from the norm.

$$d(\overline{p}, \overline{q}) = \max_i(|p_i - q_i|) = \lim_{k \to \infty} \left(\sum_{i=1}^{n} |p_i - q_i|^k \right)^{1/k} \tag{5}$$

The Canberra distance metric is used in situations where elements in the vector are always non-negative. In the case of the WIPER simulations, the output vector is composed of the number of agents in each Voronoi cell, which is always non-negative. The formula for Canberra distance is given in Equation 6. As defined, individual elements in the distance calculation could have zero for the numerator or denominator. Thus in cases where $|p_i| = |q_i|$, the element is omitted from the result.

$$d(\overline{p}, \overline{q}) = \sum_{i=1}^{n} \frac{|p_i - q_i|}{|p_i + q_i|} \tag{6}$$

4 Experimental Setup

In order to evaluate the feasibility of our approach, we present three experiments that demonstrate the effectiveness of the measures on validating agent movement

models. The first experiment uses output from a particular run of the WIPER simulation as the synthetic data that will be tested against. This output is considered a "target" simulation. For the second movement model experiment we want to examine the effectiveness of measures in ranking models over all model types.

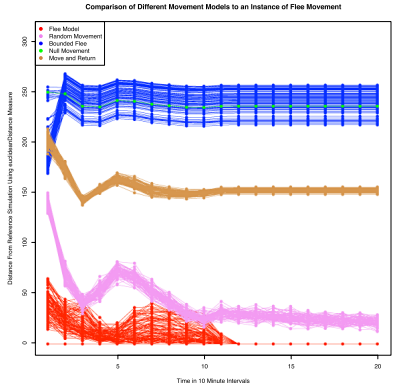
The purpose of these tests are not to demonstrate the effectiveness of the simulation to match the synthetic data but to demonstrate the ability of the measure to differentiate between simulation movement and activity model types. In a DDDAS system models of human behavior will be created and, according to the traditional model development approach, be validated offline. From this set of pre-validated models the system must be able to select, while the system is running, the model that best matches the data.

For the initial movement model experiment, we examine the effectiveness of the various statistical tests and measures in their ability to rank simulations in their closeness to the baseline simulation. The baseline simulation models a crisis scenario where people are fleeing a disaster. All simulations, including the baseline, are started with 900 agents distributed among 20 Voronoi cells. The distribution of agents to Voronoi cells is fixed over all of the simulations. For each of the 5 movement models, 100 replications of the simulation using different random seeds are run. Our evaluation approach is to examine the effectiveness of each measure in ranking output against the baseline. In this experiment, the desired results will show that instances of the Flee model are closer to the target simulation than other model types.

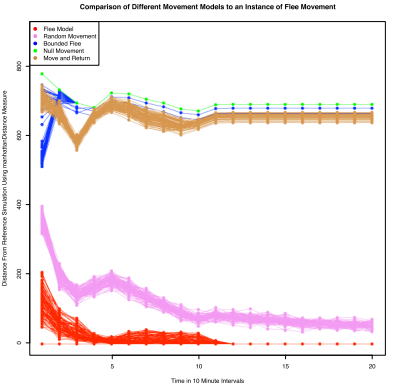
The second movement model experiment considers all of the movement models simultaneously. We use the data from the 500 simulation runs and create a matrix of distance values between every pair of values. Each position m_{ij} in the 500x500 matrix is the value of the distance metric between row i and column j. For consistency we present both the upper and lower halves of the matrix, as well as the diagonal, which is always equal to 0. We create this distance matrix for each of the distance metrics we consider. The outcome of the experiment is determined by examining the matrix and determining if the distance metric used shows low distance for simulation runs of the same model and high distance between simulation runs with differing models.

5 Results

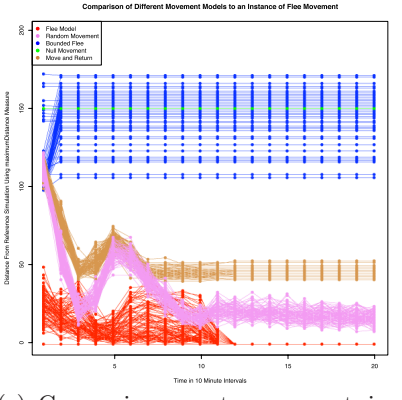
Results of using the Euclidean distance metric to measure the differences in agent locations between movement models is shown in Figure 1(a). At the first time interval the Euclidean metric does an excellent job of ranking the Flee model instances as being the best match to the baseline. All of the Flee model instances have low distance to the baseline and are all lower than any instance of the other models. Interestingly, as the simulations progress, the Euclidean distance of each simulation's output from the Flee model baseline seems to yield good results for classifying the models, as they demonstrate low inter-class distance and high intra-class distance. The exception is the Null and Bounded Flee models. This result is discussed below in the analysis.



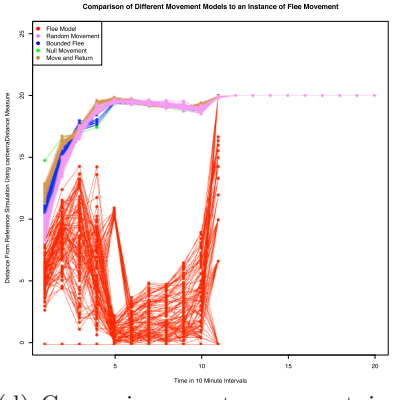
(a) Comparing agent movement in various movement models to flee movement using euclidean distance as the measure.



(b) Comparing agent movement in various movement models to flee movement using manhattan distance as the measure.



(c) Comparing agent movement in various movement models to flee movement using Chebyshev distance as the measure.

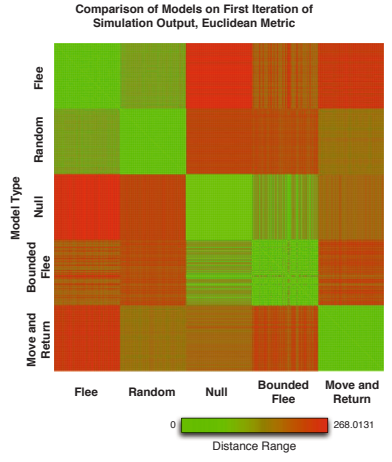


(d) Comparing agent movement in various movement models to flee movement using Canberra distance as the measure.

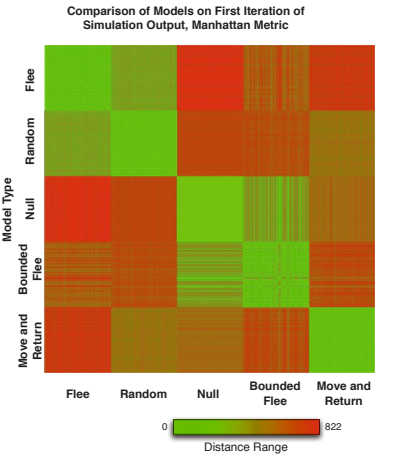
Fig. 1. Movement models compared periodically over the run of the simulations

Results of using the Manhattan distance metric to measure the differences in agent locations between movement models is shown in Figure 1(b). The Manhattan metric produces similar results to the Euclidean metric, with good results beginning at the first time interval.

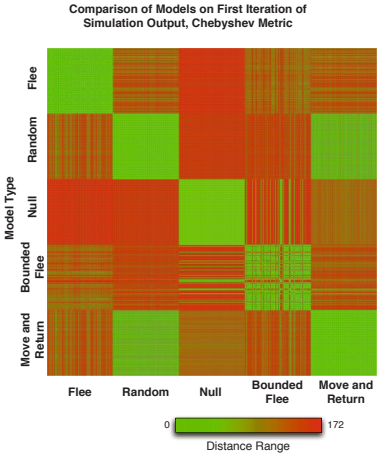
Results of using the Chebyshev distance metric to measure the differences in agent locations between movement models is shown in Figure 1(c). As with the other measures in the L family, the Chebyshev distance metric does a good job of differentiating between model types in the early stages of the simulation run and in the late stages.



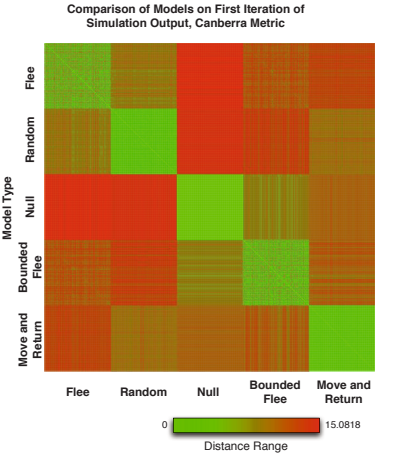
(a) Plot of the euclidean distance between simulation output. Simulations are grouped along x- and y-axis according to movement model in the order Flee movement, Random movement, Null movement, Bounded Flee movement and Move and Return movement.



(b) Plot of the manhattan distance between simulation output. Simulations are grouped along x- and y-axis according to movement model in the order Flee movement, Random movement, Null movement, Bounded Flee movement and Move and Return movement.



(c) Plot of the Chebyshev distance between simulation output. Simulations are grouped along x- and y-axis according to movement model in the order Flee movement, Random movement, Null movement, Bounded Flee movement and Move and Return movement.



(d) Plot of the canberra distance between simulation output. Simulations are grouped along x- and y-axis according to movement model in the order Flee movement, Random movement, Null movement, Bounded Flee movement and Move and Return movement.

Fig. 2. Distance plots for the simulation outputs

Results of using the Canberra distance metric to measure the differences in agent locations between movement models is shown in Figure 1(d). The Canberra metric appropriately ranks the Flee model simulations as closest, but as the simulations progress the results appear to be unstable and beyond the 11th sampling interval the Canberra metric fails to return valid values for Flee model simulations. Also, unlike the Euclidean and Manhattan metrics, the Canberra metric displays some overlap in the distance results for different model types.

Results of plotting the distance of the output from 500 simulations, 100 runs of each of the 5 movement models, is shown in Figures 2(a), 2(b), 2(c) and 2(d). These results provide a more thorough analysis of the usefulness of the metrics than simply comparing to one run of a simulation. In the ideal scenario the matrix will display low distance between simulations of the same model type (in the figures this would be a bright green square on the diagonal) and high distance when measured against simulations of a different model type (orange or red squares in the remainder of the matrix).

The figures measure the distance in the respective metric of the first time interval of the simulation. The simulations are grouped according to model type in the order, left to right (and top to bottom), Flee Movement, Random Movement, Null Movement, Bounded Flee Movement and Move and Return Movement. Each figure is colored from green to red, with green representing low distance and red high, with the colors scaled to the range of the distance values for the respective metrics.

Figures 2(a), 2(b) and 2(c) present the results for the Euclidean, Manhattan and Chebyshev metrics, respectively. Each of these metrics presents fairly good results in giving simulations of the same model type low distances and simulations with different model types high distance.

The results of the Canberra metric is less clear. The Canberra metric, Figure 2(d) appears to produce high distance values for Flee model simulations against other Flee model simulations and likewise for the Bounded Flee model.

6 Conclusions

Using a distance metric for model selection has several advantages. An experimental study, like that presented in this paper, allows users to calibrate the selection threshold, which makes it possible for the DDDAS to classify a phenomenon based on the distance from the validated model to the event. Alternately, should no model meet the threshold, the system can determine that none of the models are appropriate. In that case the measure may give a suggestion for the "closest fit" model and provides a scenario for new model creation.

In this paper we have presented an evaluation of various tests and measurements for online validation of Agent-Based Models. We have shown that the Euclidean and Manhattan distance metrics work well for validating movement models, however the Canberra distance are significantly less useful.

The Manhattan, Euclidean and Chebyshev metrics produce favorable results when used for measuring the similarity of simulations. Under the conditions we have tested, they produce low inter-model distances with high intra-model

distance. Any of these metrics is adequate for an application such as the WIPER project.

The Canberra metric is useful under certain circumstances, but the poor performance measuring Flee model simulations against other Flee model simulations make it less than desirable for use in the WIPER project.

Figures 1(a), 1(b) and 2(b) show the distance metrics failing to differentiate between the Bounded Flee model and the Null Movement model. This result is an artifact of the way movement is measured in the simulations. Since agent locations are aggregated to the level of the Voronoi cell, agent movements below this resolution do not appear in the output. In the Bounded Flee model, agents move 1000 meters from the crisis and then stop moving. Thus, if the crisis is centered in a Voronoi cell that is approximately 2000 meters across, agents in the crisis cell will not appear to have moved at all.

A caveat concerning the use of simple thresholds for model selection in online validation: In the WIPER project, where mislabeling a crisis event as benign could have dire consequences, it is important to factor into the system the cost of false negatives. Crisis event models should be weighted so that when confronted with a crisis event, the chance of labeling it as normal behavior is minimized.

7 Future Work

The work in this paper has focused on measurements for online validation of agent movement models, where validation is selection from among a set of alternatives. Agent behavior in the WIPER simulation is composed of both movement and activity models. It is important for the online validation procedure to treat both movement and activity. In the future we would like to examine measurements for online validation of agent activity models, perhaps in conjunction with work being done to characterize crisis behavior as seen in cell phone activity data [11]. In keeping with our framework, we will need to create not only different input parameters for the activity models, but new models that describe agent behavior under different scenarios (normal activity, crisis, etc). Such work on generating additional agent activity models is currently under way.

Acknowledgements

The graphs in this paper were produced with the R statistics and visualization program and the *plotrix* package for R [12][13].

References

1. Solicitation, N.P.: DDDAS: Dynamic data-driven application systems. NSF Program Solicitation NSF 05-570 (June 2005)
2. Darema, F.: Dynamic Data Driven Application Systems: A new paradigm for application simulations and measurements. In: Bubak, M., van Albada, G.D., Sloot, P.M.A., Dongarra, J. (eds.) ICCS 2004. LNCS, vol. 3038, pp. 662–669. Springer, Heidelberg (2004)

3. Schoenharl, T., Bravo, R., Madey, G.: WIPER: Leveraging the cell phone network for emergency response. International Journal of Intelligent Control and Systems 11(4) (December 2006)
4. Balci, O.: Verification, Validation, and Testing. In: Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice, John Wiley & Sons, New York (1998)
5. Kennedy, R.C.: Verification and validation of agent-based and equation-based simulations and bioinformatics computing: identifying transposable elements in the aedes aegypti genome. Master's thesis, University of Notre Dame (2006)
6. Xiang, X., Kennedy, R., Madey, G., Cabaniss, S.: Verification and validation of agent-based scientific simulation models. In: Yilmaz, L. (ed.) Proceedings of the 2005 Agent-Directed Simulation Symposium, April 2005. The Society for Modeling and Simulation International, vol. 37, pp. 47–55 (2005)
7. Davis, W.J.: On-line Simulation: Need and Evolving Research Requirements. In: Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice, John Wiley & Sons, New York (1998)
8. Andradóttir, S.: Simulation Optimization. In: Handbook of Simulation: Principles, Methodology, Advances, Applications, and Practice, John Wiley & Sons, New York (1998)
9. Wikipedia: Voronoi diagram — Wikipedia, the free encyclopedia (2006) (accessed April 25, 2006), `http://en.wikipedia.org/w/index.php?title=Voronoi_diagram&oldid=47842110`
10. Wikipedia: Chebyshev distance — wikipedia, the free encyclopedia (2007) (accessed May 14, 2007)
11. Yan, P., Schoenharl, T., Madey, G.R.: Application of markovmodulated poisson processes to anomaly detection in a cellular telephone network. Technical report, University of Notre Dame (2007)
12. R Development Core Team: R: A Language and Environment for Statistical Computing. R Foundation for Statistical Computing, Vienna, Austria (2006) ISBN 3-900051-07-0
13. Lemon, J., Bolker, B., Oom, S., Klein, E., Rowlingson, B., Wickham, H., Tyagi, A., Eterradossi, O., Grothendieck, G.: Plotrix: Various plotting functions, R package version 2.2 (2007)