

eXplainable Modeling (XM): Data Analysis for Intelligent Agents

Demonstration

Alberto Castellini
University of Verona
alberto.castellini@univr.it

Riccardo Sartea
University of Verona
riccardo.sarte@univr.it

Francesco Masillo
University of Verona
francesco.masillo@studenti.univr.it

Alessandro Farinelli
University of Verona
alessandro.farinelli@univr.it

ABSTRACT

Intelligent agents perform key tasks in several application domains by processing sensor data and taking actions that maximize reward functions based on internal models of the environment and the agent itself. In this paper we present eXplainable Modeling (XM), a Python software which supports data analysis for intelligent agents. XM enables to analyze state-models, namely models of the agent states, discovered from sensor traces by data-driven methods, and to interpret them for improved situation awareness. The main features of the tool are described through the analysis of a real case study concerning aquatic drones for water monitoring.

KEYWORDS

Data analysis; situation assessment; interpretability

ACM Reference Format:

Alberto Castellini, Francesco Masillo, Riccardo Sartea, and Alessandro Farinelli. 2019. eXplainable Modeling (XM): Data Analysis for Intelligent Agents. In *Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), Montreal, Canada, May 13–17, 2019*, IFAAMAS, 3 pages.

ONLINE MATERIAL (VIDEO)

<https://youtu.be/2qV0ixaXSrQ>

1 INTRODUCTION

Intelligent agents have many applications domains, such as, monitoring, digital manufacturing, surveillance and rescuing [11]. These systems are usually equipped with several sensors able to collect large amounts of data about the agent itself and the environment. An example is aquatic drones for autonomous water monitoring of catchments. They navigate rivers and lakes to acquire real-time data concerning important water parameters, such as dissolved oxygen and electrical conductivity [4–8, 12]. The data collected by these agents represents a valuable source of knowledge, but the extraction of this knowledge requires sophisticated data analysis [2] and statistical learning [3, 14] techniques.

In this paper we present an open source Python software called eXplainable Modeling (XM) [10], which supports the extraction of knowledge from data generated by intelligent agents. It integrates a set of recent state-of-the-art methodologies for clustering, subspace

clustering, time series segmentation, sequential data modeling and dimensionality reduction, with interactive visualization tools and statistical measures of performance/informativeness. These tools are synergistically used to generate agent's state-models from traces of sensor data, and to analyze and interpret these models, which is a key aspect for situation awareness [1, 9]. State-models here are meant as abstract, compact and informative descriptions of some properties of the agent and the environment, in specific situations faced by the agent. For instance, a state-model with very small electrical conductivity could represent the state "drone out-of-water", and a model with very high variation of heading, the state "turn". State-models are key blocks of knowledge that may be used for task planning, anomaly detection, predictive maintenance, energy management and other important purposes.

2 TOOLS FOR STATE-MODEL ANALYSIS

XM is currently able to load state-models generated by six state-of-the-art methodologies, namely, SubCMedians subspace clustering [17], Toeplitz Inverse Covariance-Based Clustering (TICC) [13], Inertial Hidden Markov Models (IHMM) [16], Gaussian Mixture Models (GMM) [3], k-means [3], and spectral clustering [3]. Moreover, it can load results of dimensionality reduction performed by Principal Component Analysis (PCA) [3] and t-Distributed Stochastic Neighbor Embedding (t-SNE) [21].

2.1 Model file format

Model file format. A key element for the integration of the statistical learning and data analysis methodologies in XM is a unified *model file format*. The following files are required by XM to load state-models generated by any methodology: *i) the dataset file*, a csv file with observations in the rows and variables in the columns; *ii) the labeling file*, a csv file containing state labels generated by the state-models for each data observation; *iii) the model parameter file*, a csv file containing all state-model parameters, i.e., cluster centroids for SubCMedians and k-means, inverse covariance matrices for TICC, means and covariance matrices for GMM and means, covariance and transition matrices for IHMM; *iv) the general information file*, a csv file containing all the information needed for model reproducibility (e.g., learning parameters) and state-models performance measures (e.g., accuracy, silhouette, BIC/AIC); *v) the standardization file* (optional), a csv file containing mean and standard deviation of each variable in the dataset; *vi) the ground truth file* (optional), a csv file containing true class labels for each data observation.

Proc. of the 18th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2019), N. Agmon, M. E. Taylor, E. Elkind, M. Veloso (eds.), May 13–17, 2019, Montreal, Canada. © 2019 International Foundation for Autonomous Agents and Multiagent Systems (www.ifaamas.org). All rights reserved.

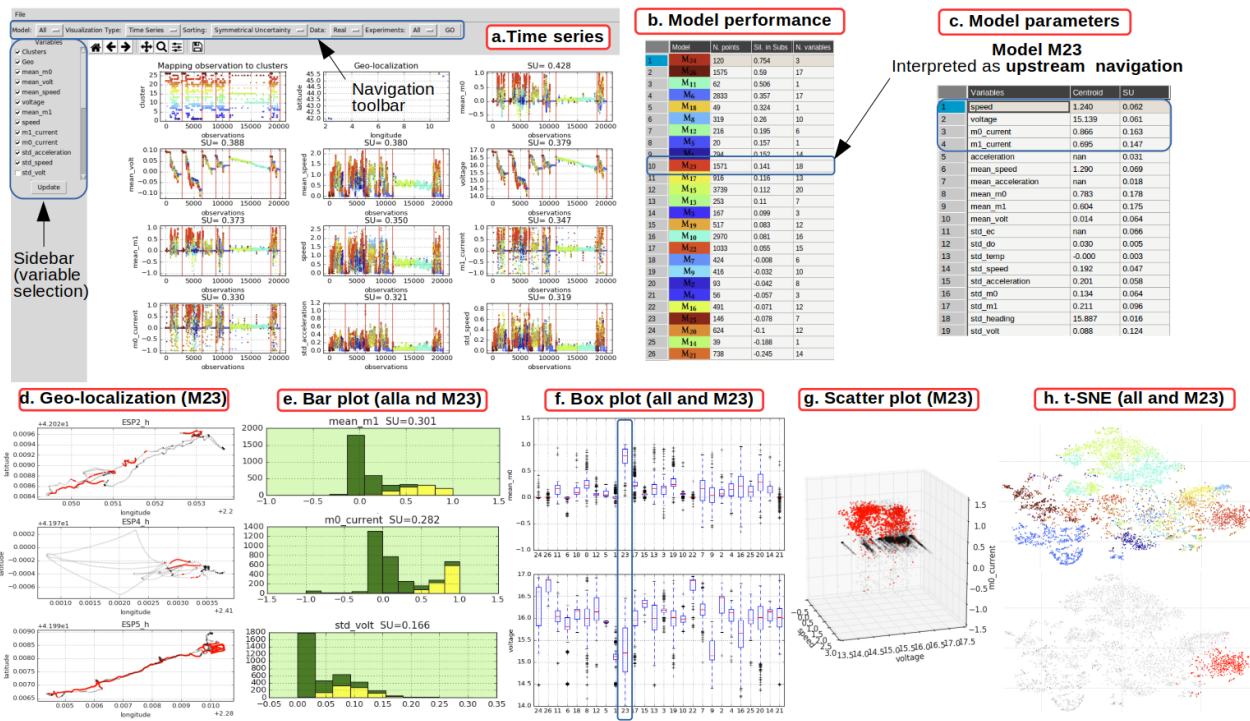


Figure 1: Visualizations available in XM. State-models (different colors) are generated from real dataset of aquatic drones.

Analysis and visualization tools. The XM graphical user interface (Fig. 1.a) has three main sections, namely, a *navigation toolbar* (on top), a *sidebar* (on the left), and a central *visualization pane*. The navigation toolbar enables to select specific models (“Model” drop-down), visualizations (“Visualization type” drop-down), experiments (“Experiment” drop-down), variable sorting strategies (“Sorting” drop-down), and to switch between original and standardized data (“Data” drop-down). The sidebar enables to select subsets of variables or subsets of experiments. The central visualization pane displays interactive charts and tables.

The strength of XM is the user-friendly and interactive integration of several types of visualization and statistical measures in the same environment (see some of them in Figure 1.a-h) that provide different perspectives on state-models and enhance their interpretability. The currently available visualization types are: *i)* general information and performance measures about state-models (Fig. 1.b), *ii)* state-model parameters (Fig. 1.c), *iii)* time series (Fig. 1.a), *iv)* geo-localization (Fig. 1.d), *v)* bar plot (Fig. 1.e), *vi)* box plot (Fig. 1.f), *vii)* scatter plot (Fig. 1.g), *ix)* PCA and t-SNE dimensional reduction (Fig. 1.h), *x)* heatmaps of dataset and states.

Figure 1 shows, as an example, the analysis of state-models generated from a real dataset collected by water drones. Model M_{23} was selected because of its particular pattern in the geo-localization view, characterized by upstream navigation in two rivers (Fig. 1.d). The analysis of model parameters, bar plots and box plots (Fig. 1.c,e,f) confirm specific properties for this state-model, such as high commands to propellers (called m_0 and m_1 current), which is typical in upstream navigation to contrast the water flow. The 3D scatter plot and t-SNE visualizations (Fig. 1.g,h) confirm that observations

of state-model M_{23} are compactly distributed. Other ways to identify interesting states is the analysis of model performance (Fig. 1.b) and the analysis of extreme distributions in boxplots (Fig. 1.f).

3 TOOLS FOR STATE-MODEL GENERATION

State-models can be generated both independently from XM, by scripts using public libraries [15, 18–20], or by the XM model generation toolbox. The first solution is more flexible but it requires a converter to generate results in the standard XM file format (see Section 2.1). The second solution requires a manual insertion of algorithm parameters by the user (a possible limit if several models must be generated) but it generates results immediately loadable by XM.

4 CONCLUSIONS

XM supports data analysis for intelligent agents through the integration of recent statistical learning and visualization tools able to generate and interpret agent state-models. Results on a real case study show the usefulness of the tool for improving situation awareness in autonomous aquatic drones, and foresee developments to other application domains, such as, anomaly detection, predictive maintenance and battery management.

ACKNOWLEDGMENTS

The research has been partially supported by the projects "Dipartimenti di Eccellenza 2018-2022, funded by the Italian Ministry of Education, Universities and Research (MIUR), and "GHOTEM/COREWOOD, POR-FESR 2014-2020", funded by Regione del Veneto.

REFERENCES

- [1] V. Berenz, F. Tanaka, and K. Suzuki. 2012. Autonomous battery management for mobile robots based on risk and gain assessment. *Artificial Intelligence Review* 37, 3 (2012), 217–237.
- [2] M. R. Berthold and D. J. Hand (Eds.). 2003. *Intelligent Data Analysis: An Introduction* (2nd ed.). Springer-Verlag, Berlin, Heidelberg.
- [3] C. M. Bishop. 2006. *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Springer-Verlag New York, Inc., Secaucus, NJ, USA.
- [4] L. Bottarelli, M. Bicego, J. Blum, and A. Farinelli. 2019. Orienteering-based informative path planning for environmental monitoring. *Engineering Applications of Artificial Intelligence* 77 (2019), 46–58.
- [5] A. Castellini, G. Beltrame, M. Bicego, D. Bloisi, J. Blum, M. Denitto, and A. Farinelli. 2018. Activity Recognition for Autonomous Water Drones based on Unsupervised Learning Methods. In *Proc. 4th Italian Workshop on Artificial Intelligence and Robotics (AI*IA 2017)*, Vol. 2054. 16–21.
- [6] A. Castellini, G. Beltrame, M. Bicego, J. Blum, M. Denitto, and A. Farinelli. 2018. Unsupervised activity recognition for autonomous water drones. In *Proceedings of the Symposium on Applied Computing, SAC 2018*. ACM, 840–842.
- [7] A. Castellini, F. Masillo, M. Bicego, D. Bloisi, J. Blum, A. Farinelli, and S. Peigner. 2019. Subspace clustering for situation assessment in aquatic drones. In *Proceedings of the Symposium on Applied Computing, SAC 2019*. ACM, To appear.
- [8] P. Duranti. 2015. *CatOne, Multitask Unmanned Surface Vessel for Hydro-Geological and Environment Surveys*. Springer, 647–652.
- [9] M. R. Endsley. 1995. Toward a Theory of Situation Awareness in Dynamic Systems. *Human Factors* 37, 1 (1995), 32–64.
- [10] eXplainable Modeling code. 2019. <https://github.com/XModeling/XM>. (2019).
- [11] A. Farinelli, D. Nardi, R. Pigliacampo, M. Rossi, and G. P. Settembre. 2012. Co-operative situation assessment in a maritime scenario. *International Journal of Intelligent Systems* 27, 5 (2012), 477–501.
- [12] G. Ferri, A. Manzi, F. Fornai, F. Ciuchi, and C. Laschi. 2015. The HydroNet ASV, a Small-Sized Autonomous Catamaran for Real-Time Monitoring of Water Quality: From Design to Missions at Sea. *IEEE Journal of Oceanic Engineering* 40, 3 (2015), 710–726.
- [13] D. Hallac, S. Bhooshan, M. Chen, K. Abida, R. Sodic, and J. Leskovec. 2018. Drive2Vec: Multiscale State-Space Embedding of Vehicular Sensor Data. In *2018 IEEE 21th International Conference on Intelligent Transportation Systems (ITSC)*. IEEE, 3233–3238.
- [14] T. Hastie, R. Tibshirani, and J. Friedman. 2001. *The elements of statistical learning*. Springer New York Inc., New York, NY, USA.
- [15] Inertial Hidden Markov Models (IHMM) code. 2015. <https://github.com/george-montanez/InertialRegularizedHMM>. (2015).
- [16] G. Montanez, S. Amizadeh, and N. Laptev. 2015. Inertial Hidden Markov Models: Modeling Change in Multivariate Time Series. (2015).
- [17] S. Peignier, C. Rigotti, A. Rossi, and G. Beslon. 2018. Weight-based search to find clusters around medians in subspaces. In *Proceedings of the Symposium on Applied Computing, SAC 2018*. ACM, 471–480.
- [18] Scikit learn code. 2019. <https://scikit-learn.org/>. (2019).
- [19] SubCMedians subspace clustering code. 2018. <https://sergiopeignier.github.io/>. (2018).
- [20] Toeplitz Inverse Covariance-Based Clustering (TICC) code. 2018. <https://github.com/davidhallac/TICC>. (2018).
- [21] L. van der Maaten and G. Hinton. 2008. Visualizing data using t-SNE. *Journal of Machine Learning Research* 9 (2008), 2579–2605.