

Monte-Carlo Based Reinforcement Learning (MCRL)

Muath Alrammal and Munir Naveed

Abstract—This paper presents a Monte-Carlo based Reinforcement Learning approach called MCRL. MCRL is applied in different domains to construct context-aware model for mobile computing. For mobile devices, we present MCRL, a light weight solution that allows to work under tight resource constraints. It is based on a multivariate regression method to construct a probability distribution which is used to construct a value function. A machine learning algorithm is applied to map the action schema with the contexts. MCRL is evaluated using different benchmark datasets. The results are compared with the start-of-the-art rivals of MCRL. The main candidate rivals of MCRL are based on the latest variations of Naves-bias, bootstrap-ping (Ada) and Decision Tree (J48). The results show that MCRL finds the quality solutions in a far shorter time than its rivals.

Index Terms—Context-awareness, regression, monte-carlo simulations, reinforcement learning, regression.

I. INTRODUCTION

Pervasive commerce (m-commerce) is known as doing commerce activities via wireless environment such as wireless Internet, and mobile handled devices which are growing very fast [1]. Currently, users use mobile phones or personal digital devices at any place and time for performing the electronic commerce activities such as e-shopping. M-Commerce is growing more and more due to easily available mobile applications for online transactions.

Computer applications are context independent. In fact, applications are based on a static and predefined picture of the context established by a system designer. Context awareness has been involved and studied in very diverse domains such as: Ubiquitous Computing, Human Machine Interaction, Information Retrieval, Recommendation Systems, Ambient Intelligence, Mobile Computing and Sensorial Networks [2]-[6].

Context awareness [7] has great potentials for creating new services and improving service quality in m-commerce. But mobile applications are less efficient in providing relevant contents to the user as compared to their web the counterparts for online shopping. Among the challenges in development of the context-aware applications for m-commerce are (i) the limited computational capability of the mobile devices and (ii) mobility. Users preferences might depend on their current location. Base on this, location is one of the main parameters in determining the current context of a mobile device.

The development and the implementation of successful context-aware applications in m-commerce needs understanding the concept of the "context" of mobile

consumers. Also, it requires to understand how to access and utilize contextual information in an appropriate way. In addition, it requires a sophisticated algorithm to build the learning model very fast for any size of dataset.

This paper introduces an approach called MCRL. It is a new one in modeling a domain to construct context-aware models for mobile computing. MCRL is based on a multivariate regression method to construct a probability density function for action schema in a domain.

The remainder of the paper is structured as follows: Section II presents the related approaches in this field. Section III describes the main contributions of this paper. Section IV presents our novel approach, named MCRL. Section V illustrates the efficiency of our model through detailed experimental results. Section VI concludes with an outline of future work in this direction.

II. RELATED WORK

In Ref. [8], the authors present a context-aware model for the mobile recommendation systems. The context-aware model is built in conjunction with a collaborative filtering process. The context semantics are predefined and the contents relevant to the current context are retrieved using a cosine similarity technique i.e. the contents that are similar (or have a minimum distance) to the current context are queried. The retrieved contents are further processed to perform collaborative filtering by using Self-Organizing Maps. The custom-built datasets are used to evaluate the context-aware model. The results show that the model (given in [8]) performs better when the number of recommendations increase in size.

In [9], the authors present a framework that includes classification of contextual aspects in a domain. The framework is based on the machine learning algorithms to retrieve the relevant aspects of a context. The machine learning algorithm is used for acquiring relevant contents which provide personalized adaptations for each learner profile. The context-aware model is composed of two main functionalities: personalization and automatic customization of contents. The classification layer of the framework is able to categorize the data into the classes belonging to the personal context and shared context. The shared context represents the com-munity based features. The combination of personal and shared context builds a personalized profile for a user and outcomes of the model are based on the profile. Bayesian networks with two different probability distributions have been explored by [10] with the focus on speeding up the generalization process.

The authors [10] in explores two stochastic processes: Dirichlet Process [11] and Gaussian Process [12] to build probability distribution in a bayesian network. The presented bayesian network has a linear time complexity with respect to

Manuscript received March 19, 2019; revised December 7, 2019.

Muath Alrammal and Munir Naveed are with the Computer Information Sciences, Higher Colleges of Technology, UAE (e-mail: malrammal@hct.ac.ae, mnaveed@hct.ac.ae).

the number of users. The user preferences are modelled as a Dirichlet Process (DP) which reduces the space complexity for the representation. The user preferences are generalized by using a Gaussian Process (GP) based bayesian network. The experiments are performed using a benchmark dataset called car preferences. The results show that the hybrid of GP and DP performs better than only GP based solution of [12].

A time-efficient decision tree construction approach is presented in [13]. Their decision tree method is called Very Fast Decision Tree Learner (VFDT). VFDT generalizes data that appears in a stream e.g. a real-time data given by sensors. The engine of VFDT is based on Hoeffding tree algorithm, which allows the method to learn in very small constant time per example. Therefore, VFDT is able to learn from massive number of examples in a very short time. However, despite the fact that VFDT is fast and cost less RAM memory while running, it is inflexible in terms of handling concepts drift, which is the hidden patterns evolve over the time in stream data. This is due to the fact that VFDT does not discard history of knowledge, it only keeps on absorbing the new data streams. However, the knowledge capture can override niche areas of domain knowledge that might be crucial for context-awareness e.g. a change in the sales trend of a high value product in a retail due to its small transaction as compared to high number of transactions of low value product.

In Ref. [14] authors concentrate on the speed of recognition as a vital parameter in the analysis of action recognition algorithms. They proposed a new algorithm for activity recognition using information from QPs (quantification parameter), MB (maroblock) partition types, and MVs (motion vectors) in the H.264/AVC compressed video. QP Gradient Image (QGI) is formed using the gradient information of QP over the space. GCI provides an evidence regarding motion occurrence and the spread of the action. The action is represented as follows: (1) Gradient Image, (2) Motion Accumulation and (3) Magnitude Weighted Orientation of Motion vectors (MWOM). Features extracted from these representations are used for classifying the actions. The features extracted are QGI Spread, QGIP (QGI projection) and MAP (motion-accumulated projection). The features extracted from QGI and MVs are used to train the PBL-McRBFN classifier to recognize the actions. In this work, the two benchmark datasets Weizmann [15] and KTH [16] were used. The recognition performance is compared with BayesNet [17] [18], SVM [19], RandomForest [20] and ELM [21], [22]. Based on the presented results, it is observed that PBL-McRBFN classifier outperforms other classifiers by more than 10%. The proposed approach can classify QCIF resolution videos with speed more than 2000 fps (frame per second) and is more than 100 times faster than existing state-of-the-art pixel domain approaches.

In Ref. [23] authors aim to develop a database tool that adequately manage data imperfection considering contextual data derived from different sources. Their architecture proposed to develop Context-Aware Fuzzy Databases is based on two main foundations: Fuzzy Databases and Zadeh's Computational Theory of Perceptions (CTP) [24]. Their proposal is derived from the contribution presented in [25].

Their fuzzy object-oriented data model is derived from [26]

[27]. They use fuzzy domains to manage attribute values with imperfect data, and they design diverse fuzzy operators (methods) for each type of domain (classes) that enable the management of these data, such as a Fuzzy Equal method (FEQ) that could be computed based on different criteria. This allows to extend the native data types and to define methods using an ORDBMS.

The context (such as user, time, location, device type or any other) is seen as any external feature to the database that is significant to the interaction between the user and the database. They classify context information as the following (1) Explicit: the context is provided by the user, both previously through a user profile or at the moment of performing the interaction with the database. (2) Implicit: the context is inferred from data provided by environmental sensors or software agents categories. The use of the proposed architecture allows to model reality more adequately (through the representation of imperfect data), represent the perceptions of people (using linguistic labels), and emulate human natural processes (as communication among people).

DCPE co-training [28] is a hybrid approach used for data classification. This approach integrates various learning algorithms including Naive Bayes, neural networks, k-nearest neighbors, and decision tree. The approach designed for binary classification, it uses two different classifiers to predict better labeled data to be assigned to unlabeled data. DCPE co-trainings promote diversity in both classifiers which results in improved accuracy. However, DCPE approach needs to be tested in domains with large number of input features to validate the effectiveness of this approach.

Monte-Carlo (MC) Simulations are used for power scheduling in [29] where demand forecasting is performed by using a fuzzy logic-based estimation model. MC simulations are run using a data-driven based generative model that can simulate the stochastic behavior of demand and supply process in power scheduling. MC simulator estimates "Loss Of Load Expectation (LOLE)". An error function - based on difference between actual LOLE and estimated LOLE - is used to measure performance of MC simulator. MC simulator is compared to a four-layered Neural Network for the same dataset. The results show that performance of MC simulator is comparable to the trained neural network.

Ensemble of classifiers have been explored in [30] to solve the Human Activity Recognition (HAR) problem. Performance of classifier ensemble is determined by using accuracy, area under curve and F-measure. The result shows that Ensemble of classifiers perform better than Multi-layered Neural Network, logistic regression and C4.5 (decision tree) in accuracy. However, Multi-Layered Neural Network performs better to classify observations better than other models if the observation belongs to either "jogging" or "standing" classes.

MOCART [31], [32] uses MONTE-CARLO simulations to approximate the probability distribution for action selection in a context model. MOCART builds the action schema using an unsupervised learning-based approach called MCPROD. MCPROD estimates the probability distribution for predicting the action transitions by using a tweak of Monte-Carlo Tree Search. Each context is represented as a

state and a change in context is modeled as a state-transition. The pre-processing of the raw data involves the construction of state-automata. MOCART is empirically investigated using the benchmark dataset (*i.e.* Car Preference dataset [10]). The results show a clear advantage of using MOCART for context-awareness over a variety of rivals e.g. state-of-the-art variations of Decision Tree and Naive Bayes.

A hybrid of memory-based and model-based approaches have been explored in [34] to classify the preference-based dataset. In the hybrid approach, K-means is used to construct clusters of the preferences and then a collaborative filtering is applied on the refined clusters using a voting-based ranking method. The collaborative filter (CF) approach is used to predict the preference of a user. The CF algorithm picks all the relevant instances of a given user's preferences from a cluster and ignores the irrelevant from the same cluster. The hybrid approach is compared against three other techniques: GroupLens CF [34] (GCF), a correlation-based CF (CCF) and k-means based CF (KCF). Three evaluation parameters are used to compare the results which are size of response set, size of search space and t-value. The results show the hybrid is a better than its rival however it is not scalable with respect to size of item set.

A ranking based CF is presented by Kamishima *et al.* [35] to predict the user preferences. The ranking method is applied on two model-based Collaborative Filtering algorithms: pLSA [36] and Matrix Decomposition [37]. The experiments are performed using Sushi dataset. pLSA and Matrix Decomposition are trained exclusively using ranking based approach and scored based filtering approach. The results show that ranking-based approach improves the prediction accuracy for the dataset if the number of choices in the preferences are more than two.

In Ref. [38] a data mining (DM) approach is proposed to predict the success of telemarketing calls for selling bank long-term deposits. This approach seeks to increase profits and reduce costs. The data from 2008 to 2013 was collected from a Portuguese bank. The dataset is unbalanced, as only 6557 (12.38%) records are related with successes. For evaluation purposes, a time ordered split was initially performed, where the records were divided into training (four years). A semi-automatic feature selection was explored in the modeling phase, performed with the data prior to July 2012 and that allowed to select a reduced set of 22 features and test data (one year). Four binary classification DM models were implemented in the rminer package of the R tool: logistic regression (LR), decision trees (DTs), neural network (NN) and support vector machine (SVM). Two metrics, they are: area of the receiver operating characteristic curve (AUC) and area of the LIFT cumulative curve (ALIFT) were used to test the four models. As a result, the NN outperformed the other models. It had the best results (AUC = 0.8 and ALIFT = 0.7), allowing to reach 79% of the subscribers by selecting the half better classified clients.

III. MOTIVATIONS AND CONTRIBUTIONS

The behavior of the mobile applications is entirely characterized by using the current context of the application and the mobile device. The notion of context can vary from application to application, however, a simple and common

form of context can be defined in terms of location and user preference. The user preferences can vary with a change in the location. For example, a user of a shopping app would prefer to browse a different set of contents at home as compared to the use of the same app at office

One of the essential elements of the commercial mobile applications is to predict the user preferences (particularly within a context) to provide a better usage experience for their customers. Such a prediction model is also required for consumer behavior modelling. The size and dimensions of the set of contexts for a mobile application can grow exponentially when the number of users increase. These characteristics make the context-awareness modelling a challenging and non-trivial task.

It is quite intuitive to use a probabilistic model to solve context-awareness problem. The most of the best performing methods for context-aware modelling are probabilistic e.g. Naive Bayes [10], [28], Monte-Carlo Simulations [32] and Decision Tree [39].

In this work, we express context-awareness as a preference prediction problem. The model is learning in an anytime algorithm to meet the resource constraints. The anytime algorithm can stop at any given instance to the meet the tight time constraints and returns a near-optimal solution. However, it completes its processing if time-constraints are flexible and returns optimal solution.

Our contributions are as the following:

- Developing a machine learning algorithm called MCRL. It is used to solve feature selection and prediction problems. MCRL exploits the strengths of Monte-Carlo simulations and Regression under the tight resource constraints.
- Providing an example to explain the simulation process of MCRL.
- Testing MCRL using the two benchmark datasets, which are: Bank [38] and Wine [40]. Providing a comparison of MCRL with four state-of-the-art best performing prediction algorithms.

IV. MCRL

This section describes the basic terminologies and algorithmic details of Monte-Carlo based Reinforcement Learning (MCRL). Furthermore, we provide an example that explains how MCRL functions.

A. Problem Definition

Definition 1: F is a finite set of domain features.

Definition 2: L is a finite set of class labels that can be characterized by elements of F .

Definition 3: C is combination of features such that $C \subseteq F$.

Definition 4: $T: C \times C \rightarrow L$ is a functional mapping between $c \in C$ and $l \in L$. T is built by MCRL.

Definition 5: $P(L|c)$ represents the probability density function for label $l \in L$ with respect to a combination $c \in C$.

Definition 6: $R: L \times C \rightarrow L$ represents regression coefficient of a linear regression between set of labels L and features combination C .

The probability of occurring of each $l \in L$ is calculated by applying a non-linear regression mapping function R . the

function R is updated such that it tunes learning structure T can decide a label for each set of features.

Example:

Suppose we have a training dataset of four observations, as given in table I. There are four combinations of features $S = \{C_1, C_2, C_3\}$ and the corresponding labels are $L = \{l_1, l_2, l_1, l_2\}$. The combination elements of C are the set of values of the feature's parameters which are $C_1 = \{1, 2\}$, $C_2 = \{1, 3, 4, 6\}$ and $C_3 = \{3, 4, 7\}$.

TABLE I: EXAMPLE DATASET

Sr. No	C ₁	C ₂	C ₃	L
1	1	3	3	l_1
2	1	4	3	l_2
3	1	6	7	l_1
4	2	1	4	l_2

MCRL executes the probability function P for each combination according to the corresponding each label. The T function is built using the probability distribution and Regression values as shown in Table II. These values are tuned by using value-function. The values represent the probability for occurrence of class label l for set of given parameters in C .

TABLE II: T FUNCTION UPDATES

T functional parameters	Values
$T(C_1, l_1)$	0.0032
$T(C_2, l_1)$	0.0145
$T(C_1, l_2)$	0.323
$T(C_2, l_2)$	0.595

B. Algorithmic Details

A high level of algorithmic details of MCRL are given in Fig. 1. MCRL applies a learning threshold called t . MCRL loads the training examples in memory as shown in Line 1. It randomly initializes the R function weights and then initiates the R structure. The model iteratively learns the weights and updates R function. The process for R updates is repeated several times until the estimated error is less than a predefined threshold t .

Procedure MCRL(S)

1. $Y = \text{LoadTrainingObservation}()$
2. $W = \text{getInitialWeights}()$
3. $R(\text{InitialMappingFunction})$
4. REPEAT
5. FOREACH y in Y
6. $x = R(Y, W)$
7. $E = \text{CalculateError}(y, x)$
8. $\text{UpdateWeight}(E, R)$
9. If $E < t$ EXIT

Fig. 1. High level algorithmic details of MCRL.

MCRL determines the preferences by using the weight vector and Tree structure of R . The selection method is trivial as it uses the preference with the highest functional value for an observation. The R updates depend on error space. The higher the error, the larger the number of simulations is run.

The learning of the $\text{updateweight}R()$, Line 8 Figure is based on Monte-Carlo simulation model which applies sampling techniques given in MoCART [32].

The Monte-Carlo simulations in MCRL tests the value function using structure R with the samples it generates from training dataset. If the latest change in the model parameters

produce higher error as compared to its pre-simulation error, then more samples are added to validate the change and the number of simulations is increased. If error rate does not reduce then changes are undone. The simulations perform look ahead for all weight updates during the training phase.

Simulations uses the random walk to generate samples for validation purposes. The probabilities values are not updated during simulations. The value function builds the R structure in the form of decision tree. The depth of the tree structure is tuned by probabilities values as shown in Table 2. The value of a leave node does not change for consecutive k iterations, and its probability value is smaller than C , then this branch is pruned from the structure. The value of k is identified during simulations such that it does not increase the error rate.

The simulation model works like an anytime search algorithm. It works with fixed number of simulations but can extend its search if additional simulations are required for validating a change in value function.

V. EXPERIMENTAL RESULT

A. Experimental Setup

Experiments run on Windows server 2012, Intel(R) Xeon(R) CPU E5-2630 v2 2.6 GHz, RAM 32 GB.

MCRL was simulated using Visual studio 2010, C # language. WEKA 3.7 was used to compare our model with the related techniques (Nayes-bias, boot-strapping (Ada) and Decision Tree (J48)). The learning model of each technique was validated using 10 folds (k folds). Accuracy, precision and F-measure values were measured. In MCRL, the average values of 10 simulations were considered for comparison.

The following benchmark datasets were used in the experiments.

B. Datasets

Technical details of used datasets are in Table III.

Bank dataset [38] is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed. The classification goal is to predict if the client will subscribe (yes/no) a term deposit (variable y).

In Wine [40] two datasets were created, using red and white wine samples. Each expert graded the wine quality between 0 (very bad) and 10 (very excellent). Several data mining methods were applied to model these datasets under a regression approach. These datasets can be viewed as classification or regression tasks. The classes are ordered and not balanced (e.g. there are much more normal wines than excellent or poor ones). Number of Instances: red wine - 1599; white wine - 4898. We considered one attribute that is quality (score between 0 and 10).

TABLE III: DATASETS CHARACTERISTICS

Dataset	Number of observations	Number of preferences
Wine	6497	1
Bank	41188	1

C. Comparison with the Related Work

MCRL generalizes better than competitors in Bank dataset.

Fig. 2 demonstrates a comparison between MCRL and its rivals in terms of accuracy. MCRL performs better than all of its rival with respect to accuracy. Fig. 3 shows performance of all of the algorithms with respect to F-Score in Bank dataset. MCRL performs with lower F-Score than its rival mainly due to presence of majority of input features as categorical. These features limit the exploration capability of the Monte-Carlo sampling approach. MCRL exploits the local minima in the case of categorical input features and imbalances the exploration and exploitation trade-off offered by Monte-Carlo sampling. Fig. 4 shows all algorithms have similar performance as Fig. 3 in terms of precision on bank dataset for the similar reason already explained for F-Score.

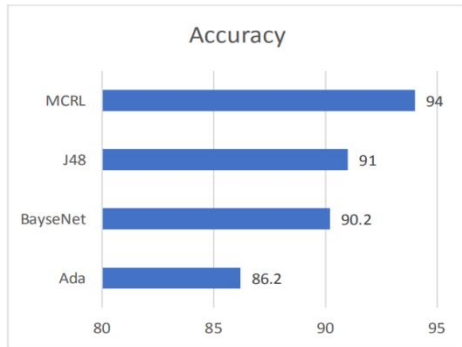


Fig. 2. Accuracy of bank dataset.

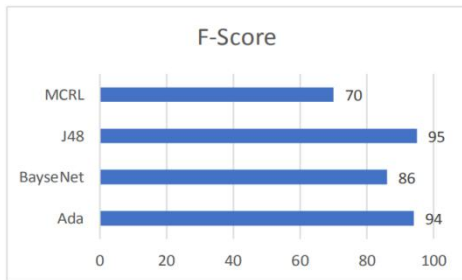


Fig. 3. F-Score of bank dataset.

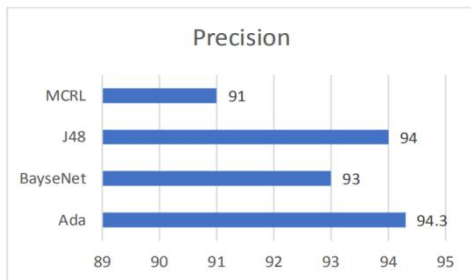


Fig. 4. Precision of bank dataset.

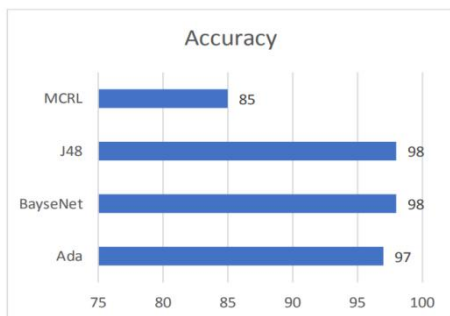


Fig. 5. Accuracy of wine dataset.

In the wine dataset, the rival techniques perform comparatively better than MCRL. Fig. 5 shows the accuracy of algorithms in wine dataset and Fig. 6 and Fig. 7 display the

F-Score and precision respectively of all algorithms in Wine dataset. These results are consistent with the results obtained from Bank dataset.

The range of values in input features are narrow which limit the capability of sampling approach. These features make it ideal for entropy-based algorithms to generalize the recommendation task easily as compared to sampling-based approach MCRL. Intuitively, the sample does not seem appropriate in a domain with input features having few numbers of possible values to occur.

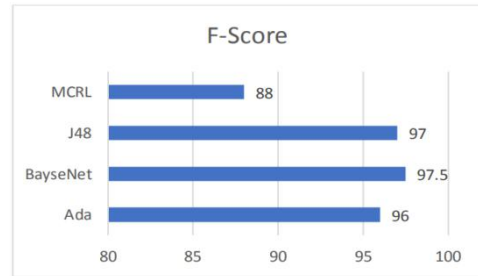


Fig. 6. F-Score of wine dataset.

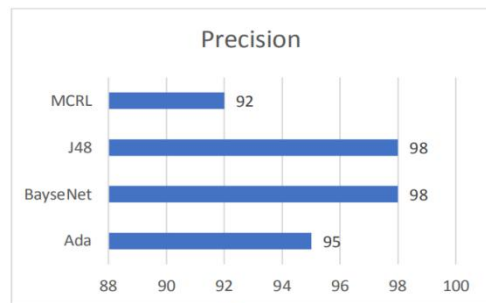


Fig. 7. Precision of wine dataset.

VI. CONCLUSION

MCRL is presented and empirically evaluated using two benchmark datasets. MCRL is based on reinforcement learning algorithm that exploits the capabilities of Monte-Carlo simulations to tune the value function. However, MCRL can converge to local-minima in the search spaces with small range of values in an observation dataset. This is particularly visible in results from Bank and Wine dataset. The results also reveal that MCRL does not suffer from over-fitting problems when other state-of-the-techniques can easily trap into it.

In future work, we will test MCRL if it can easily scalable for large and high dimensional datasets such as [41]. Also, MCRL can be modified by adding the pre-processing capabilities to identify the spaces where it can be trapped into local-minima. MCRL can also be extended to work in collaboration with other learning algorithms in the cases where search space is comprised of low dimensions and input features are dominant by categorical values.

CONFLICT OF INTEREST

Muath Alrammal and Munir Naveed declare that they have no conflict of interest.

AUTHOR CONTRIBUTIONS

In this paper, the first author Dr. Alrammal proposed the research idea, prepared the stated of the art, and proposed the

MCRL model. While the second author Dr. Naveed developed the model and conducted the experimentations. Both authors wrote this paper.

REFERENCES

- [1] T. Efraim and K. David, *E-Commerce: Mobile Commerce*, Prentice Hall, Inc., 2003.
- [2] G. Orsi and L. Tanca, "Context modelling and context-aware querying," *Lecture Notes in Computer Science*, pp. 225–244, Springer Berlin Heidelberg, 2011.
- [3] D. Martinenghi and R. Torlone, "Querying context-aware databases," *Flexible Query Answering Systems*, pp. 76–87, Springer Berlin Heidelberg, 2009.
- [4] J. Y. Hong, E. H. Suh, and S. J. Kim, "Context-aware systems," *Expert Syst. Appl.*, vol. 36, no. 4, pp. 8509–8522, May 2009.
- [5] C. Bolchini, C. A. Curino, E. Quintarelli, F. A. Schreiber, and L. Tanca, "A data-oriented survey of context models," *SIGMOD Rec.*, vol. 36, no. 4, pp. 19–26, Dec. 2007.
- [6] M. Baldauf, S. Dustdar, and F. Rosenberg, "A survey on context aware systems," *Int. J. Ad Hoc Ubiquitous Comput.*, vol. 2, no. 4, pp. 263–277, 2007.
- [7] S. Jun and M. S. Poole, "Context-awareness in mobile commerce," *Encyclopedia of Multimedia Technology and Networking*, 2005.
- [8] M. Hosseini-Pozveh, M. A. Nematbakhsh, and N. Movahhedinia, "A multidimensional approach for context-aware recommendation in mobile commerce," *CoRR*, 2009.
- [9] A.-H. Ahmed, S. Jun, and Y. Jun, "A machine learning based framework for adaptive mobile learning," in *Proc. the 8th International Conference on Advances in Web Based Learning*, Berlin, Heidelberg, 2009, pp. 34–43.
- [10] E. Abbasnejad, S. Sanner, E. Bonilla, and P. Poupard, "Learning community-based preferences via dirichlet process mixtures of gaussian processes," in *Proc. the 23rd International Joint Conference on Artificial Intelligence*, Beijing, China, August 3-9, 2013.
- [11] P. Muller and F. A. Quintana, "Nonparametric bayesian data analysis," *Statist. Sci.*, vol. 19, no. 1, pp. 95–110, 2004.
- [12] S. Guo, S. Sanner, and E. V. Bonilla, "Gaussian process preference elicitation," *Advances in Neural Information Processing Systems*, vol. 23, pp. 262–270, 2010.
- [13] P. Domingos and G. Hulten, "Mining high-speed data streams," in *Proc. the Sixth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, New York, NY, USA, 2000, pp. 71–80.
- [14] R. V. Babu, B. Rangarajan, S. Sundaram, and M. Tom, "Human action recognition in H.264/AVC compressed domain using meta-cognitive radial basis function network," *Applied Soft Computing*, vol. 36, pp. 218–227, 2015.
- [15] M. Blank, L. Gorelick, E. Shechtman, M. Irani, and R. Basri, "Actions as space-time shapes," in *Proc. Tenth IEEE International Conference on Computer Vision*, 2005.
- [16] C. Schuld, I. Laptev, and B. Caputo, "Recognizing human actions: A local svm approach," in *Proc. the Pattern Recognition, 17th International Conference on IEEE Computer Society*, USA, 2004.
- [17] G. F. Cooper and E. Herskovits, "A bayesian method for constructing bayesian belief networks from databases," in *Proc. the Seventh Conference on Uncertainty in Artificial Intelligence*, 1991.
- [18] G. Cooper and E. Herskovits, "A bayesian method for the induction of probabilistic networks from data," *Machine Learning*, vol. 9, no. 4, pp. 309–347, 1992.
- [19] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, 2011.
- [20] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, 2001.
- [21] G. B. Huang, "An insight into extreme learning machines: Random neurons, random features and kernels," *Cognitive Computation*, vol. 6, no. 3, pp. 376–390, 2014.
- [22] R. Savitha, S. Suresh, and H. Kim, "A meta-cognitive learning algorithm for an extreme learning machine classifier," *Cognitive Computation*, vol. 6, no. 2, pp. 253–263, 2014.
- [23] J. Cadenas, N. Marin, and M. Vila, "Context-aware fuzzy databases," *Applied Soft Computing*, vol. 25, pp. 215–233, 2014.
- [24] L. Zadeh, "A new direction in AI - Toward a computational theory of perceptions," *Computational Intelligence, Theory and Applications*, 2001.
- [25] J. Cadenas, N. Marin, and M. Vila, "Fuzzy domains with adaptable semantics in an object-relational dbms," *Flexible Query Answering Systems*, Springer Berlin Heidelberg, 2011.
- [26] F. Berzal, N. Marin, O. Pons, and M. Vila, "Development of applications with fuzzy objects in modern programming platforms," *International Journal of Intelligent Systems*, vol. 20, no. 11, pp. 1117–1136, 2005.
- [27] L. Cuevas, N. Marin, O. Pons, and M. Vil, "Pg4db: A fuzzy object-relational system," *Fuzzy Sets and Systems*, vol. 159, no. 12, pp. 1500–1514, 2008.
- [28] X. H. He and H. Man, "Dcpe co-training for classification," *Neurocomput.*, vol. 86, no. 75–85, June 2012.
- [29] H. Haroonabadia and M. Haghifamb, "Generation reliability assessment in power markets using monte carlo simulation and soft computing," *Applied Soft Computing*, vol. 11, pp. 5292–5298, 2011.
- [30] C. Catal, S. Tufekci, E. Pirmitt, and G. Kocabag, "On the use of ensemble of classifiers for accelerometer-based activity recognition," *Applied Soft Computing*, 2015.
- [31] M. Naveed, "Online learning based contextual model for mobility prediction," *Evolving Ambient Intelligence*, 2013.
- [32] M. Naveed, D. Kitchin, A. Crampton, L. Chrapa, and P. Gregory, "A monte-carlo path planner for dynamic and partially observable environments," in *Proc. the 2012 IEEE Conference on Computational Intelligence and Games*, 2012.
- [33] T. Kamishima, "Nantonac collaborative filtering: Recommendation based on order response," in *Proc. International Conference on Knowledge Discovery and Data Mining*, 2003, pp. 583–588.
- [34] O. Luaces, G. Bayon, J. Quevedo, J. Diez, and J. D. Coz, "Analyzing sensory data using non-linear preference learning with feature subset selection," in *Proc. the 15th European Conference on Machine Learning*, 2004, pp. 286–297.
- [35] T. Kamishima and S. Akaho, "Nantonac collaborative filtering: A model-based approach," in *Proc. the Fourth ACM Conference on Recommender Systems*, Sept. 2010, pp. 273–276.
- [36] T. Hofmann and J. Puzicha, "Latent class models for collaborative filtering," in *Proc. the 16th International Joint Conference on Artificial Intelligence*, 1999, pp. 688–699.
- [37] J. Koren, "Collaborative filtering with temporal dynamics," in *Proc. the 15th International Conference on Data Discovery and Data Mining*, 2009, pp. 447–455.
- [38] S. Moro, P. Cortez, and P. Rita, "A data-driven approach to predict the success of bank telemarketing," *Decision Support Systems*, vol. 62, pp. 22–31, 2014.
- [39] T. Tam Van Nguyen, W. Lim, H. A. Nguyen, and D. Choi, "Context awareness framework based on contextual graph," in *Proc. 5th IFIP International Conference on Wireless and Optical Communications Networks*, 2008.
- [40] P. Cortez, A. Cerdeira, F. Almeida, T. Matos, and J. Reis, "Modeling wine preferences by data mining from physicochemical properties," *Decision Support Systems*, vol. 47, no. 4, pp. 547–553, 2009.
- [41] Youtube Dataset. [Online]. Available: <https://archive.ics.uci.edu/ml/datasets/YouTube+Comedy+Slam+Preference+Data>

Copyright © 2020 by the authors. This is an open access article distributed under the Creative Commons Attribution License which permits unrestricted use, distribution, and reproduction in any medium, provided the original work is properly cited ([CC BY 4.0](https://creativecommons.org/licenses/by/4.0/)).



Muath Alrammal received his MSc in information technology from Telecom SudParis, Evry, France, in 2007. He received his Ph.D. degrees in computer sciences from the University of Paris Est, Paris, in 2011.

In 2011, he joined LACL, University of Paris Est, as a post-doctoral researcher. In 2012 he joined LIFO, University of Orleans, France, as a post-doctoral researcher. Between 2013 and 2017, he joined the IT Department in KIC, Abu Dhabi, UAE, where he was an assistant professor. Since 2017 he works for HCT, Abu Dhabi, UAE as an assistant professor in CIS Department. His current research interests include processing big data in streaming, performance models, selectivity estimation techniques, and Machine learning. Dr. Alrammal is a member of LaMHA research group.



Munir Naveed obtained his first degree in software engineering and a PhD in automated planning for real-time-strategy games. He has been exploring AI in different domains eg. computer games, big data and network security since 2014. His focus of his research is designing new algorithms to solve problems for real-time applications.