A Hybrid Active Learning and Progressive Sampling Algorithm

Amr ElRafey and Janusz Wojtusiak

***Abstract*—Sampling techniques for data mining applications can be broadly categorized into Random Sampling (RS), Active Learning (AL) and Progressive Sampling (PS). Progressive Sampling techniques grow an initial sample up to the point beyond which model accuracy no longer significantly improves. These methods have been shown to be computationally efficient. The sampling schedule to be used with progressive sampling techniques is still an ongoing issue of research due to the fact that available sampling schemes may either overshoot, resulting in a final sample which is larger than necessary, or they may grow the sample too slowly thus requiring many iterations of the algorithm before convergence is reached. We demonstrate how using Batch Mode Uncertainty Sampling from the domain of active learning, to progressively grow the sample, can significantly improve the performance of progressive sampling. Through a series of trials on both simulated and real data, we show that our proposed Progressive Batch Mode Uncertainty Sampling (PBMUS) algorithm converges with a comparable or smaller number of data points at higher accuracy and in some cases, less computational time.**

***Index Terms*—Active learning, uncertainty sampling, progressive sampling, linear regression with local sampling, random sampling, sampling, machine learning.**

I. Introduction

In recent years, the amount of data being generated every day around the world has reached staggering proportions [1]. Novel throughput technologies capable of gathering and storing vast amounts of data have ushered in the era of "Big Data" and these advancements have had a profound impact on fields as diverse as finance, government and science. The growth in the size of data sets however, has, in many cases, made standard statistical techniques and algorithms computationally infeasible and over the last decade a growing body of literature has emerged, concerned with scaling-down sampling techniques.

The task of generating a sample from a given data set is confronted by two, often competing requirements. Firstly, we wish to produce a sample which is nearly as *informative* as the entire data set. What this means, is that a learning algorithm should be able to extract the same information from the sample as it would have done, from the entire data set [2]. The second requirement of the sample is that it should be as small as possible.

Broadly speaking, scaling-down sampling techniques can be categorized into 3 types. 1) Random Sampling (RS) [2], 2) Active Learning (AL) [3] and 3) Progressive Sampling (PS) [4]. Each of these categories tackles the issue of sampling in a different way. Random sampling techniques are the easiest to implement but make no attempt to ensure that the sample drawn is as informative as the entire data set nor do they attempt to find the smallest sample possible [5]. Active learning techniques are predominantly concerned with finding the most informative data points to include in the sample [3] with the final size of the sample usually being arbitrarily decided by the user.

Progressive sampling techniques attempt to satisfy both sampling requirements by making use of the concept of the learning curve [6]. Essentially, PS techniques work by training a learning algorithm on an initial sample then gradually growing this sample until the learners accuracy no longer improves. PS is both simple to use and has been shown to be extremely efficient at generating informative samples [7].

An ongoing area of research in PS is the sampling schedule to be used, that is, the method by which new data points are added at each iteration. The most cited methods for growing the sample are Arithmetic Sampling AS [8] and Geometric Sampling GS [9]. The first of these techniques, AS, may require a very large number of iterations before accuracy converges and the second technique, GS, may *overshoot* and produce a final sample which is larger than necessary [5].

In the following paper, we demonstrate how combining ideas from the domains of progressive sampling and active learning, results in a simple and computationally tractable algorithm capable of rapidly achieving convergence. Our proposed Progressive Batch Mode Uncertainty Sampling (PBMUS) algorithm, follows the basic steps of standard PS whereby the sample is grown progressively and a convergence test is performed at each iteration. However, at each iteration of the algorithm, new data points are added using Uncertainty Sampling (US).

Results on both simulated and real data confirm that the entire learning curve is shifted to the left and convergence occurs with a comparable or smaller number of data points as PS and often at higher levels of accuracy and less computational time. Our paper begins with an overview of PS and US, followed by a description of our proposed algorithm and empirical results.

II. Overview of Current Sampling Methods

A. Progressive Sampling

The central idea of progressive sampling techniques is the *learning* curve, depicted in Fig. 1. Simply speaking, as the size of the sample grows, so too does the accuracy of any learning algorithm trained using this sample. Gradually however, the gains in accuracy become smaller for each progressive data point added to the sample until eventually,

Manuscript received June 5, 2018; revised August 18, 2018.

The authors are with George Mason University, Fairfax, VA 22030, USA (e-mail: aelrafey@gmu.edu, jwojtusi@gmu.edu).

doi: 10.18178/ijmlc.2018.8.5.723

the learners' accuracy plateaus. Progressive sampling techniques therefore, attempt to grow samples only up to the point at which accuracy plateaus and by doing so these techniques attempt to satisfy both the requirements of generating the most informative sample which is also as small as possible.

The two fundamental practical issues which arise in progressive sampling techniques are

1) *Testing for convergence of accuracy.*
2) *The sampling schedule to be used.*

Testing for convergence refers to any method of determining whether or not accuracy has converged (whether or not the sample accuracy has plateaued). As such, convergence tests essentially represent a stopping criteria and are of paramount importance. The most common technique found in the literature is *linear regression with local sampling method (LRLS)* [9].

The second practical issue of progressive sampling is the sampling schedule to be used. This refers to the sampling method used to progressively grow the sample. The issue here is that if samples are grown too rapidly, we may *overshoot* with our sample, resulting in a sample which is larger than necessary. Conversely, if samples are grown too slowly (in small increments) then the computational cost of testing for convergence at each step may be too large. The most common techniques found in the literature are arithmetic sampling (AS) [5] and geometric sampling (GS) [5] which are both variations of random sampling.

In the AS framework, an initial sample S_0 of predetermined size is selected from the data set. Subsequently, a set number of data points N_θ is consecutively added to this initial sample such that

$$S_i = S_0 + (i * N_\theta) \quad (1)$$

A simple example of arithmetically generated progressive samples where $S_0 = 500$ and $N_\theta = 100$, is $\{500, 600, 700, \ldots, 10000\}$. A major drawback of this technique is that if N_θ is too small, a very large number of iterations will be needed to reach convergence.

Alternatively, in the GS framework, the initial sample S_0 is grown geometrically by multiples of a predefined number θ such that

$$S_i = \theta^i * S_0 \quad (2)$$

A simple example of a geometrically generated progressive samples where $S_0 = 500$ and $\theta = 2$, is $\{500, 2000, 4000, \ldots, 16000\}$. With GS, a major drawback of the technique is its tendency to overshoot, since the sample size rapidly grows.

B. Active Learning

Active learning is a form of semi-supervised machine learning, in which the learning algorithm is allowed to choose the data from which it learns. At their core, active learning techniques attempt to locate the most informative data points to include in a sample. Active learning techniques can be categorized into

1) Uncertainty Sampling [10]
2) Minimizing Hypothesis Space [11], [12]
3) Variance Reduction [13], [14]

The algorithm proposed in this paper combines uncertainty sampling with PS and as such we only provide an overview of uncertainty sampling here.

Uncertainty sampling is the most commonly used and extensively researched active learning framework [15]. The basic idea is to provide a learning algorithm with an initial sample S_0 and build an initial model M_0. Subsequently, the algorithm passes through the remaining data selecting only those data points about which it is least certain to include in the sample. A simple example of uncertainty sampling would be the scenario where we have a data set containing $X_1, X_2, \ldots, X_n$ independent variables and a dependent, binary variable Y and we are interested in building a standard logistic regression model for predicting Y.

We would begin here with an initial sample S_0 and construct our initial logistic regression model M_0 which takes the form

$$Y_i = \frac{e^{(B_0 + B_1 X_{1_i} + \cdots + B_n X_{n_i})}}{e^{(B_0 + B_1 X_{1_i} + \cdots + B_n X_{n_i})} + 1} \quad (3)$$

where the $B_0, B_1, \ldots, B_i$ in the above equation are the standard logistic regression coefficients. We then pass through the subsequent data points in the full data set using the model M_0 to predict the probability of $Y = 1$ for each of the observations. Since we are dealing with a binary classifier it suffices to select the points which are closest to the 0.5 boundary [3], that is, we only select observations for which the following holds

$$0.45 \leq \frac{e^{(B_0 + B_1 X_{1_i} + \cdots + B_n X_{n_i})}}{e^{(B_0 + B_1 X_{1_i} + \cdots + B_n X_{n_i})} + 1} \leq 0.55 \quad (4)$$

Every time a new data point is added to our initial sample S_0, we use the updated sample to learn a new model M_i and use the new model to predict subsequent data points. We continue in this fashion until our sample has reached a predefined size S_{Final}. The three most common types of uncertainty found in the literature are *least confident*, *margin* and *entropy*, with the latter being the most commonly used [3].

Despite the widespread use of uncertainty sampling, a major drawback of the technique (and most other active learning techniques) is the fact that the model M is updated one instance at a time which ultimately causes computation time to increase drastically. A number of authors [16], [17] have proposed a slight modification of standard uncertainty sampling known as *batch mode uncertainty sampling*. In this framework, instead of updating the model M one instance at a time, the model is updated after a batch or group of instances have been added to the sample and the technique has been found to be very effective and less computationally costly.

III. Progressive Batch Mode Uncertainty Sampling

Again we assume the scenario where we have a data set containing $X_1, X_2, \ldots, X_n$ independent variables and a dependent, binary variable Y and we are interested in building a classification model for predicting Y. Our proposed method proceeds by selecting an initial sample S_0, applying a learner to it and constructing an initial model M_0. Using M_0, we predict the probabilities of the subsequent

$N_{Predict}$ data points and for each observation i in $N_{Predict}$ we calculate

$$\left| P_{M_0}(Y_i = 1 \mid X_{1_i}, X_{2_i}, \ldots . X_{N_i}) - 0.5 \right| \quad (5)$$

That is, we calculate the absolute distance between the predicted probability of observation i and the 0.5 decision boundary for each i in $N_{Predict}$. Having done so, we select a subgroup $N_{Uncertain}$, which are closest to the decision boundary and add them to our initial sample.

A new model M_1 is then learnt and we test our updated sample S_1 for convergence. If convergence is detected, our algorithm ends, if not we repeat the process of adding $N_{Uncertain}$ from subsequent points until convergence is detected.

Essentially, our proposed algorithm is a modification of PS with arithmetic sampling, only the N_θ data points from equation (1) above are selected using uncertainty sampling. Furthermore, we do not update our model M after each individual observation is added but only after $N_{Uncertain}$ points have been added, that is, we only update the model M at each iteration of the algorithm, as would be the case in standard PS. At each updating of the model, we test for convergence of the algorithm and we use the test most commonly found in the literature which is LRLS.

We expect our proposed method to deliver a number of advantages to standard PS or US including

1) Convergence should occur with a smaller number of data points than standard PS due to the fact that PBMUS selects only those points about which the model M is least certain about. Specifically, we expect the learning curve depicted in Fig. 1 to be shifted to the left.
2) Due to the fact that PBMUS will pass through a greater number of data points, but only select a few, this technique is not prone to ignoring large clusters of data which may not be located near the beginning of the data set.
3) The computational cost of using US is greatly reduced, since we only update the model M at each iteration of the algorithm and not after each data point is added.

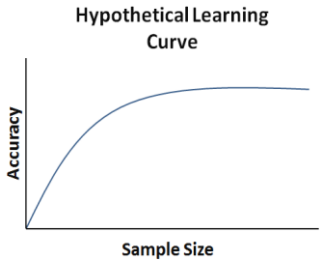

Fig. 1. Hypothetical learning curve.

IV. Simulation Results

To begin with, we created a data set with 30 explanatory binary variables and one binary outcome variable and sampled the coefficients of the explanatory variables from the uniform distribution with $\{\min = -4, \max = 4\}$. We created two million training observations and 200,000 testing observations. We used Logistic Regression as a learning algorithm with an initial sample of 500 observations for both PBMUS and PS and set, $N_\theta = N_{Uncertain} = 100$. That is, at each iteration of both methods we added 100 observations where the PS algorithm added these observations randomly and PBMUS added them using uncertainty sampling. $N_{Predict}$ was set at 10,000 , meaning that at each iteration of our algorithm, we use our model M_i to predict the subsequent 10,000 data points selecting only the closest 100 points to 0.5 decision boundary.

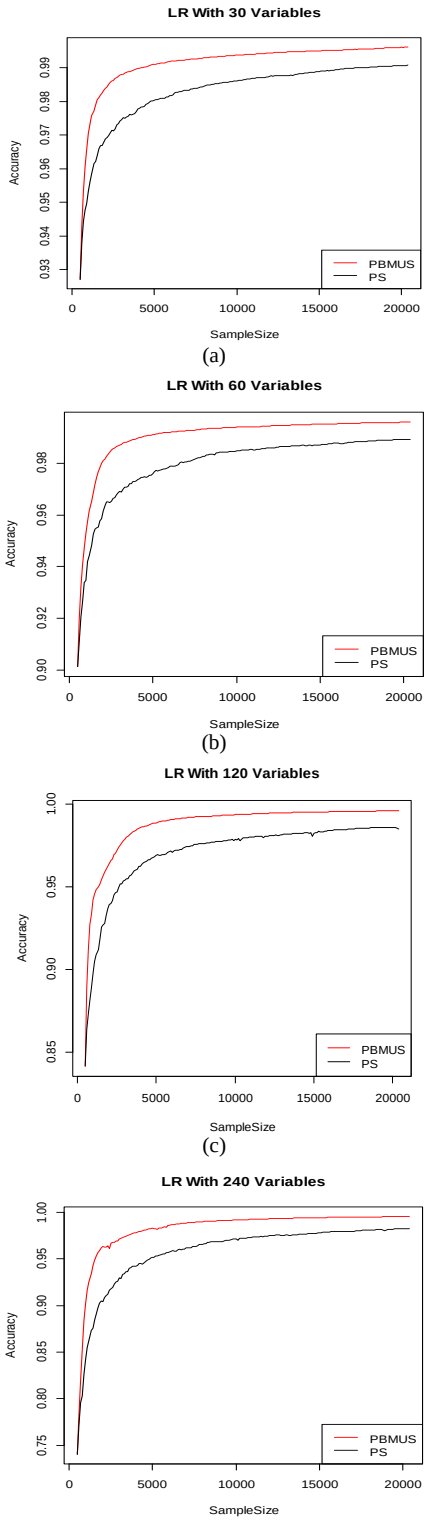

Fig. 2. Simulation results.

This simulation was repeated 100 times, with a new synthetic data set each time and the results on data sets with 30 covariates are shown in Fig. 2(a). We furthermore repeated the same simulations on synthetic data sets with 60, 120 and 240 explanatory variables, sampling the coefficients

of the explanatory variables from the uniform distribution with {min = −4 , max = 4 } and the results are all displayed in Fig. 2.

A summary of our simulation results are presented in Table I below. The results indicate that PBMUS managed to reach convergence with a fewer number of data points and at a higher accuracy and as would be expected, the difference between PBMUS and PS is larger in data sets with a larger number of variables. It is also worth noting that the accuracy of PBMUS remained stable as the number of explanatory variables increased whereas the accuracy of PS gradually declined as the number of variables increased.

A more remarkable result however concerns the computational time required to perform both algorithms. In the data sets with 30 explanatory variables, PBMUS required an additional 15 seconds to reach convergence as compared to PS. However, as the number of explanatory variables increased, the computational time of PS increased at a faster rate and eventually, PBMUS was actually faster.

TABLE I: COMPARISON OF PBMUS AND PS APPROACHES

	PS Convergence	PBMUS Convergence	PS Accuracy	PBMUS Accuracy	Average Computational Time PS	Average Computational Time PBMUS
30 Variables	7100	5600	98.30%	99.15%	30	45
60 Variables	9600	6000	98.45%	99.19%	75	84
120 Variables	10100	7400	97.80%	99.20%	117	144
240 Variables	12800	8900	97.50%	99.07%	482	400

V. TEST ON REAL DATA

We further tested our proposed technique on real data sets from the UCI Repository.

A. Poker Hand Data

The Poker Hand data set [18] contains 10 explanatory variables and 1 outcome variable with 1 million training data points and 25,010 testing observations. The outcome variable is not binary as was the case in our simulations, but is composed of 10 different categories. Furthermore, for this data set we used C5.0 [19] as our learning algorithm instead of LR.

As was the case with our simulated data we set the initial sample S_0 at 500 observations but given the relatively limited number of training observations available we set $N_{Predict} = 10{,}000$ and $N_{Uncertain} = 5{,}000$. At each iteration of the algorithm, we selected the closest 50% of predicted observations to the decision boundary as opposed to the closest 1% as was the case in our simulation studies. Fig 3 below depicts the results.

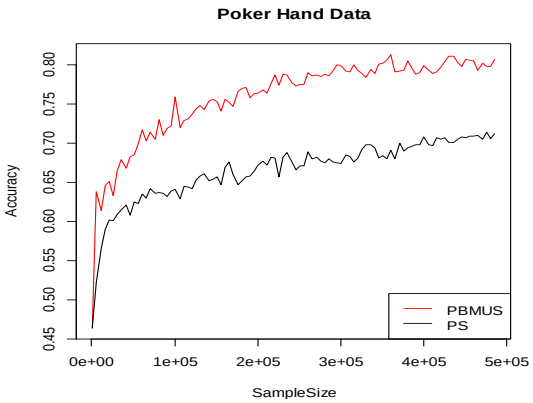

Fig. 3. Results on Poker Hand data set.

Our results are depicted in Fig. 3. As was the case in our simulations studies, the sample generated by PBMUS dominates the PS sample in terms of accuracy. Specifically, convergence with PBMUS occurred after 349 seconds at a sample size of 316,000 data points with an accuracy of 79.9% while with PS convergence occurred after 558 seconds at a sample size of 416,000 data points and an accuracy of 70.7%.

B. Credit Default Data

The credit default data set [18] contains 23 explanatory variables and 1 binary outcome variable with a total of 30,000 observations. For this data set, we used 25,000 observations as training data and 5,000 observations as testing data. Furthermore, we used standard logistic regression as our learning algorithm and the Area Under the Curve statistic (AUC) to compare the 2 techniques instead of accuracy. We set S_0 at 100, $N_{Predict} = 200$ and $N_{Uncertain} = 100$. That is, at each iteration of the algorithm, we again selected the closest 50% of predicted observations to the decision boundary. Fig. 4 below depicts the results.

As is clear in Fig. 4, the learning curve generated by PBMUS is shifted to the left and dominates the PS learning curve. Convergence with PBMUS occurred with 3,900 observations at an AUC of 73.4% and with PS it occurred with 7,900 observations at an AUC of 73.2%.

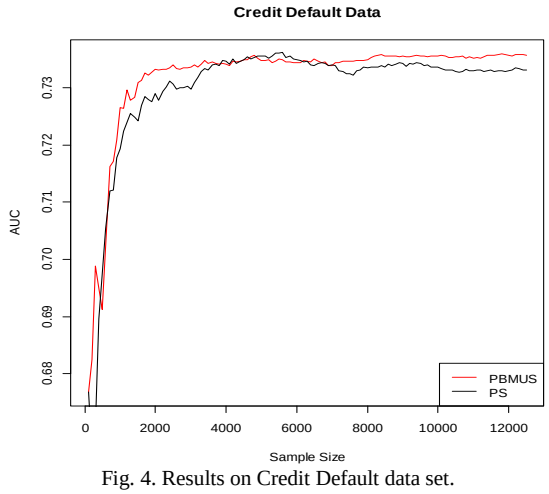

Fig. 4. Results on Credit Default data set.

C. Wine Quality Data

The wine quality data set [18] contains 11 explanatory variables and 1 outcome variable representing wine quality taking values from 0 to 10. We recoded the outcome variable as follows:

- If wine quality <= 5 then wine quality = 0
- If wine quality > 5 then wine quality = 1

There are a total of 6,497 data points (for both white wine and red wine) and as such we used 5,000 observations as training data and 1,497 observations as testing data. For this data set, we used Logistic Regression and included all second order interaction terms in the models and the AUC statistic was again used to compare the performance of the sampling techniques. Given the size of the small size of the data we set S_0 at 50, $N_{Predict} = 20$ and $N_{Uncertain} = 10$ and the results are depicted in Fig. 5 below.

Using LRLS, convergence with PBMUS was detected after 980 data points were added to the sample at an AUC of 80% whereas with PS convergence was detected with a sample of 930 data points at an AUC of 79.2%. So although PBMUS did not generate a smaller sample at convergence, it did generate a sample of very similar size and with greater AUC. It is also worth noting here, that LRLS detected convergence fairly early for both PBMUS and PS since we continued growing the samples beyond convergence and a greater AUC was achieved with both sampling techniques.

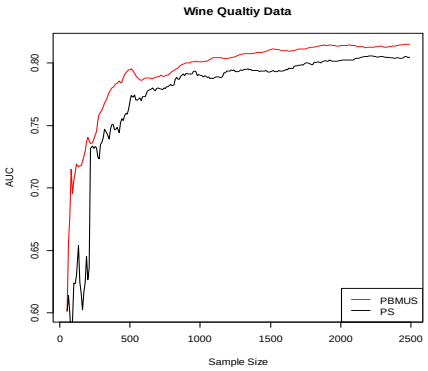

Fig. 5. Results on Wine Quality data set.

VI. Conclusion

In a survey conducted of annotation projects [17] for natural language processing tasks, it was found that only 20% of respondents had ever decided to use active learning techniques and one of the main reasons cited for this lack of interest was the time required to execute most available strategies. In this paper, we have attempted to combine elements of both progressive sampling and active learning to develop a technique that is both effective and computationally tractable.

Notwithstanding many of the shortcomings of progressive sampling, the fundamental concept of using a learning curve to grow samples, is intuitively sound and has been demonstrated to be very efficient in many cases. By combining this concept with a simple modification of standard uncertainty sampling, we have demonstrated that it is possible to generate samples which are smaller in size, produce greater accuracy and require little additional or even less computation time.

References

[1] *IBM What Is Big Data: Bring Big Data to the Enterprise*. (2012). [Online]. Available: http://www-01.ibm.com/software/data/bigdata/
[2] L. Huan and H. Motoda, "Instance selection and construction for data mining," *Springer Science & Business Media*, vol. 608, 2013.
[3] B. Settles, 2012, "Active learning," *Synth Lect Artif Intell Mach Learn*, vol. 6, no. 1, pp. 1–114.
[4] C. Meek, B. Theisson, and D. Heckerrnan, "The learning-curve sampling method applied to model- based clustering," *The Journal of Machine Learning Research*, 2002.
[5] E. Amr and J. Wojtusiak, "Recent advances in scaling-down sampling methods in machine learning," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 9, no. 6, 2017.
[6] M. Christopher, B. Thiesson, and D. Heckerman, "The learning-curve sampling method applied to model-based clustering," *Journal of Machine Learning Research*, vol. 2, pp. 397-418, Feb. 2002.
[7] B. H. Gu, B. Liu, F. F. Hu, and H. Liu, "Efficiently determining the starting sample size for progressive sampling," in *Proc. European Conference on Machine Learning*, Springer, Berlin, Heidelberg, 2001, pp. 192-202.
[8] G. H. John and P. Langley, "Static versus dynamic sampling for data mining," *KDD*, vol. 96, pp. 367-370. 1996.
[9] P. Foster, D. Jensen, and T. Oates. "Efficient progressive sampling," in *Proc. the Fifth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, ACM, 1999, pp. 23-32.
[10] D. D. Lewis and J. Catlett, "Heterogeneous uncertainty sampling for supervised learning," in *Proc. the Eleventh International Conference on Machine Learning*, 1994, pp. 148-156.
[11] T. M. Mitchell, "Generalization as search," *Artificial Intelligence*, vol. 18, no. 2, pp. 203-226, 1982.
[12] C. David, L. Atlas, and R. Ladner, "Improving generalization with active learning," *Machine Learning*, vol. 15, no. 2, pp. 201-221, 1994.
[13] C. Kathryn and I. Verdinelli, "Bayesian experimental design: A review," *Statistical Science*, pp. 273-304, 1995.
[14] F. Valerii, "Optimal experimental design," *Wiley Interdisciplinary Reviews: Computational Statistics*, vol. 2, no. 5, pp. 581-589, 2010.
[15] S. Manali and M. Bilgic., "Evidence-based uncertainty sampling for active learning," *Data Mining and Knowledge Discovery*, vol. 31, no. 1, pp. 164-202, 2017.
[16] C. Rita, Z. Wang, W. Fan, I. Davidson, S. Panchanathan, and J. P. Ye, "Batch mode active sampling based on marginal probability distribution matching," *ACM Transactions on Knowledge Discovery from Data (TKDD)*, vol. 7, no. 3, 2013.
[17] P. Swarnajyoti and L. Bruzzone, "A batch-mode active learning technique based on multiple uncertainty for SVM classifier," *IEEE Geoscience and Remote Sensing Letters*, vol. 9, no. 3, pp. 497-501, 2012.
[18] D. Dua and E. K. Taniskidou. (2017). *UCI Machine Learning Repository*. [Online]. Available: http://archive.ics.uci.edu/ml
[19] Q. J. Ross, *C4. 5: Programs for Machine Learning*, Elsevier, 2014.

Amr ElRafey obtained a master of arts degree in statistics from Columbia University in 2016 and is currently a graduate research assistant and PhD candidate in knowledge discovery and health informatics at George Mason University. His research interests include causal inference in health data, sampling methods and feature selection techniques for large data sets.

Janusz Wojtusiak is an associate professor of health informatics and director of the Machine Learning and Inference Laboratory, George Mason University. His research expertise includes machine learning, health informatics, artificial intelligence in clinical decision support and knowledge discovery in medical data, machine learning, evolutionary computation, intelligent evolutionary design, knowledge mining, health data analytics, and a wide range of applications of these fields in health care. His particular area of interest is in developing algorithms that derive simple and transparent models from complex healthcare data.

He authored or co-authored over 70 research publications and presentations and continues to collaborate with multiple national and international institutions including University of Louisville, University of Bremen, Germany, and AGH University of Science and Technology. In 2007, he was a fellow of Hanse-Wissenschaftskolleg (Hanse Institute for Advanced Study), Germany and was a post-doctoral fellow at Mason after completing his Ph.D.