SAND2015-3239C

Repeated Play of the SVM Game as a Means of Adaptive Classification

Craig M. Vineyard, Stephen J. Verzi, Conrad D. James and James B. Aimone
Sandia National Laboratories
Albuquerque, New Mexico 87185-1327
Email: cmviney@sandia.gov

Gregory L. Heileman
Electrical and Computer Engineering Department
University of New Mexico
Albuquerque, NM 87131-0001

***Abstract*—The field of machine learning strives to develop algorithms that, through learning, lead to generalization; that is, the ability of a machine to perform a task that it was not explicitly trained for. An added challenge arises when the problem domain is dynamic or non-stationary with the data distributions or categorizations changing over time. This phenomenon is known as concept drift. Game-theoretic algorithms are often iterative by nature, consisting of repeated game play rather than a single interaction. Effectively, rather than requiring extensive retraining to update a learning model, a game-theoretic approach can adjust strategies as a novel approach to concept drift. In this paper we present a variant of our Support Vector Machine (SVM) Game classifier which may be used in an adaptive manner with repeated play to address concept drift, and show results of applying this algorithm to synthetic as well as real data.**

I. Introduction

Machine learning was originally defined in 1959 by Arthur Samuel as a "field of study that gives computers the ability to learn without being explicitly programmed [1]." With this goal in mind, a wide variety of techniques have been developed such as artificial neural networks, clustering algorithms, and statistical techniques. Many of these approaches are analogous to the subfield of mathematics that is differential equations. Unlike calculus in which the properties of a known function are analyzed to find regions of interest such as extrema, with differential equations the function relating the variables of interest is unknown and various techniques are employed to find a function which describes the dynamics of the problem domain. Machine learning algorithms likewise seek to infer a solution, whether that be neural network weights or distribution priors, such that the algorithm can perform tasks like classification for example, without having been given the fully known (and thus explicitly programmable) algorithm up front. Machine learning has had great success in a wide variety of applications such as voice recognition, computer vision, and signal processing.

A key limitation to these approaches arises when the data provided for the machine learning algorithm to learn from is dynamic or non-stationary. There are numerous scenarios under which the data generating process a machine learning algorithm is attempting to learn is non-stationary. This may be due to changes in the data over time, known as concept drift [2]. There are two primary categories of concept drift, virtual and real. In a virtual concept drift setting, changes occur in the underlying data distribution feature space (which may drive a need to update the model) [3]. This is also termed sampling shift [4]. For example, in comparing automobile statistics over time fuel efficiency and weight are two metrics which have changed historically with advances in technologies, variability in materials used, as well as consumer interest at a given time period (i.e., whether sport utility vehicles or sports cars are in demand). In the presence of virtual concept drift, it is possible for the learned decision boundary to change over time even though labeling of previously seen samples do not change. It may also be the case that not all of the data is available *a priori* (whether due to recording limitations or other availability constraints). Several examples of virtual concept drift will be used to demonstrate the utility of the game theoretic classifier described in this paper. Semi-Supervised Learning (SSL) algorithms are another approach for addressing data availability constraints where unlabeled data is used in conjunction with labeled data for training a classifier [5]. The limited labeled data available *a priori* is used to make inferences about the additional unlabeled data, and both are subsequently used to train a classifier. SSL algorithms are similar to incremental learning in the sense that through iterations data is refined as the classification model is updated. Alternatively, in the real concept drift paradigm (also known as concept shift), the concepts themselves are changing and this may be irrespective of the underlying data distribution [3]. Or in other words, real concept drift is a change in the class space where data points may change their class membership over time. As an example, buyer preference models evolve as consumers watch more media or attain more products. Furthermore, the rate at which concept drift occurs may be characterized broadly as sudden or gradual.

There are three general approaches to addressing concept drift: instance selection, instance weighting, and ensemble learning [6]. Instance selection approaches define a window over which the model is applicable. Instance weighting approaches assign significance to data by various means, and take this significance into account regarding how or if models are updated. And finally, ensemble learning approaches construct multiple models which are combined (typically by weighting or voting) to yield the model output [6]. Depending upon the problem at hand and the approach employed, it may not always be feasible to generate a new machine learning model as the data is updated or changed. Doing so may necessitate retraining a neural network and may be a slow, computationally intensive procedure. Rather, it is desirable to be able to adapt and update models so they may be continued to be used.

In the following sections we will give an overview of how the mathematics of game theory addresses the problem of dynamic opponent strategies and changing preferences through repeated games. We then present our Support Vector Machine (SVM) Game classifier and describe how a variant may be used in an adaptive manner with repeated play to address virtual concept drift as an instance weighting approach. We then show results of applying this adaptive algorithm to synthetic as well as real data.

II. Repeated Play in Game Theory

Game theory is a branch of applied mathematics used to formally analyze the strategic interactions between competing players [7][8][9]. As such, game theory has been applied to a wide variety of real world domains such as international relations, business and military tactics, as well as auctions and elections [8]. Machine learning problems encapsulate various forms of conflicting interactions. Classification tasks, for example, are based upon the premise that a decision must be made to choose how a given input should be classified. While to many this may not seem like a canonical domain for game theory, it is quite amenable in the sense that a strategic decision must be made, and data points or classifiers (depending upon how the problem is cast) can be shown to behave rationally. Decision boundaries are often optimized to minimize classification error and in that sense are competing against one another to be the resulting discriminant. Data points which provide the basis upon which a classifier is trained influence the shape, orientation, or position of discriminant boundaries.

Consequently, in a concept drift situation this implies the classifier must be able to adapt accordingly. In game theory, this is analogous to players adjusting their strategies over time. Rather than repeatedly playing a losing, or suboptimal, strategy it is often the case that players will adjust their play (assuming they are not in an equilibrium setting where unilaterally changing their behavior cannot improve their expected payoff). Rather than a single shot game in which the players interact only once, repeated games examine long term interactions. In this sense, players take into account the effect their actions may have on opponent's future behavior. Repeated games may be finitely repeated or infinitely repeated. The notion of taking into account long term interactions applies to machine learning classifiers also because it is desirable to have a classifier which generalizes well (as opposed to any arbitrary discriminant which correctly classifies the data used for training). And so, as follows we will describe our game-theoretic SVM Game classifier first as a finitely repeated game, and then discuss how as an infinitely repeated game it applies to the concept drift paradigm.

III. SVM Game

The SVM approach to learning, seeks to find the separating hyperplane that maximizes the margin between the patterns in the classes it is separating, and these patterns serve as the support vectors [10][11]. Conceptually, this is similar to taking into account the long term classification goal as opposed to settling for the first discriminant which yields no training error. Furthermore, Bennett et al. proved there is a geometric interpretation which is equivalent to the dual of the canonical quadratic optimization approach to SVM [12]. This approach first constructs the convex hulls, the smallest convex set of points which fully encompass the set, around each of the data classes. Next it finds the closest points to each-other on each respective convex hull. The resulting discriminant is the perpendicular bisector of the line segment formed by these points. Figure 1 illustrates both the fundamental SVM principle as well as the geometric SVM approach. In the top half of the figure, the green rectangle represents the margin between the classes and the resulting discriminant is the black line central to this region. The bottom half of the figure depicts the identical discriminant resulting from the geometric approach.

With a desired outcome or a goal in mind, game theoretic mechanism design develops a framework defining player actions and the effect of these actions in efforts to attain the desired goal [13]. Using the geometric SVM learning paradigm as a desired goal, we have developed an iterated game to identify which data patterns are closest to the opposing class and thus define the position and shape of the resulting discriminant. Our SVM Game is a two player iterated game where the data patterns are the players. The Condorcet Method is a technique for aggregating the preferences of multiple voters and determining a resulting decision by taking into account pairwise comparisons. A Condorcet winner beats every other candidate in pairwise comparison [14]. As a Condorcet method, our game evaluates pairwise interactions between data points. Each iteration of the game randomly selects two players from the same class and one data pattern from the opposing class. The pattern from the opposing class is not a player in the game, but rather provides a reference to determine which player is closer to the opposing class [15]. In canonical SVM, an alpha value (α) is a scalar multiplier of the support vectors. All data points initially start with the same finite amount of α, and through optimization the α is redistributed to the support vectors in amounts corresponding to their influence on the discriminant. Likewise, in our game, each player (data point) starts with an initial (equal) quantity of α. For each iteration of the game, competing players pass or hold a percentage of their α. Individual players do not choose which action to take (pass or hold), but rather their actions are dictated by their proximity to the reference point from the opposing class. In this sense, rather than players choosing a strategy, their actions correspond to innate properties of the players [16]. Fig. 2 shows the basic SVM Game algorithm just described.

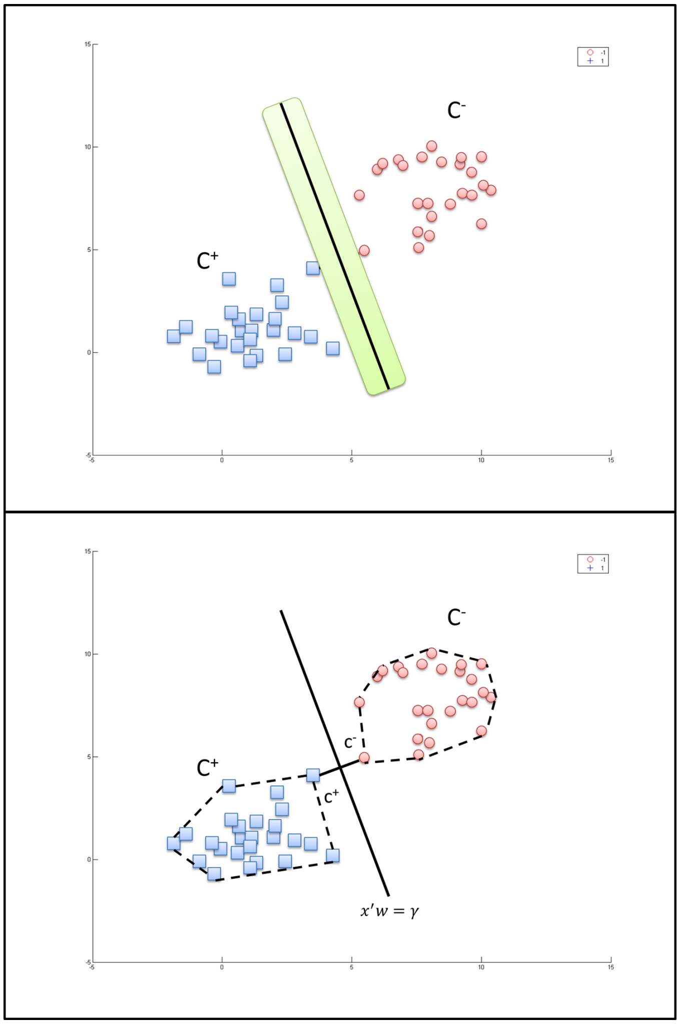

Fig. 1. Support Vector Machine Maximum Margin Principle and Equivalent Geometric SVM

Algorithm 1 Basic SVM Game

procedure SVM_GAME$_B(S)$ ▷ Sets $C^+, C^- \in S$
  while iterations≠desired iterations **do**
    for each class C^+ and C^- **do**
      $p_x \leftarrow random(C^1)$
      $p_y \leftarrow random(C^1)$
      $p_r \leftarrow random(C^2)$
      if $\mathrm{d}(p_x, p_r) \leq \mathrm{d}(p_y, p_r)$ **then**
        $p_x \leftarrow \alpha$ from p_y
      else
        $p_y \leftarrow \alpha$ from p_x
      end if
    end for
  end while
  return (p_i, p_j)
    s.t. $d(p_i, p_j) < d(p_m, p_n)\ \forall\ m, n \neq i, j$ and $p_{i,m} \in C^+, p_{j,n} \in C^-$
end procedure

Fig. 2. Basic SVM Game Algorithm

Additionally, as an extension to the basic SVM Game, a coalitional SVM Game provides stability as well as a means of addressing non-linear problems. In this game variant, each player (data point) has a coalition partner which is an affiliation of a data point with a single member of the opposite class believed to be the closest member of the opposing class. Coalition partners are one-way pairings which may be many to one. Effectively, this builds coalitions within a given class of the grouping of like-minded players who all agree upon the preferred (closest) player of the opposing class. Every iteration of the coalitional SVM game allows each interacting player to consider the relative distance to both the reference point from the opposing class as well as their coalition partner. When all players in a given class form the same coalition, they are in agreement as to which player is the closest point amongst the opposing class and this unanimous Condorcet winner allows a linear discriminant to be constructed if both classes form single coalitions. If a unanimous decision cannot be reached this illustrates that a Condorcet winner does not exist and rather a non-linear solution is needed. In lieu of an unanimous Condorcet winner, rather the irreducible coalitions constitute Smith Sets which are a partitioning of the global problem such that within each of the Smith sets there is a local Condorcet winner [17]. Since each Smith Set consists of a local unanimous Condorcet winner, a global non-linear solution may be constructed by the composition of these local solutions. The coalitional SVM Game algorithm is shown in Fig. 3, and for more details regarding the SVM Game see [18].

Algorithm 2 Coalitional SVM Game

procedure SVM_GAME$_C(S)$ ▷ Sets $C^+, C^- \in S$
  for all $p_i \in S$ **do** ▷ Initialization
    $p_i \leftarrow$ Initial Coalition Parter from Opposite Class
  end for
  while iterations≠desired iterations **do**
    for each class C^+ and C^- **do**
      $p_x \leftarrow random(C^1)$
      $p_y \leftarrow random(C^1)$
      $p_r \leftarrow random(C^2)$
      Coalition$(p_x) \leftarrow \min(\mathrm{d}(p_x, p_r), \mathrm{d}(p_x, Coalitiona(p_x)), \mathrm{d}(p_x, Coalition(p_y)))$
      Coalition$(p_y) \leftarrow \min(\mathrm{d}(p_y, p_r), \mathrm{d}(p_y, Coalitiona(p_y)), \mathrm{d}(p_y, Coalition(p_x)))$
      if $\mathrm{d}(p_x$,Coalition$(p_x)) \leq \mathrm{d}(p_y$,Coalition$(p_y))$ **then**
        $p_x \leftarrow \alpha$ from p_y
      else
        $p_y \leftarrow \alpha$ from p_x
      end if
    end for
  end while
  return (p_i, p_j)
    s.t. $d(p_i, p_j) < d(p_m, p_n)\ \forall\ m, n \neq i, j$ and $p_{i,m} \in C^+, p_{j,n} \in C^-$
end procedure

Fig. 3. Coalitional SVM Game Algorithm

The SVM Game is a class of algorithms, and we have presented two versions (Basic and Coalitional) here. The training phase consists of the selection of players and reference point for a game iteration. The SVM Game class of algorithms employs online learning since it only evaluates a single player-player-reference triple at a time (as opposed to taking into consideration all interactions simultaneously). For a static data set, a batch training variant of the SVM Game may be used such that an epoch comprised of desired player-player-reference triples is selected *a priori*. The SVM Game would still iterate upon a single player-player-reference triple at a time, but the set of triples would be fixed. Doing so allows bounds to be placed on the number of game iterations needed to converge to a solution. Namely, in the worst case all players evaluate every opposing class point as a reference. Such a Brute-Force SVM Game requires $\mathcal{O}(n^2)$ game iterations. A fixed epoch approach is analogous to a finitely repeated game. However, just as infinitely repeated games allow for the emergence of different game dynamics than finitely repeated games, by playing the SVM Game in an online training manner allows the SVM Game to address non-static data. Rather than formulating a fixed epoch of all triples to iterate over, by selecting these one at a time the SVM Game employs online training. As follows we present results with an online training paradigm.

IV. Results

To demonstrate the merits of repeatedly playing the SVM Game in a concept drift paradigm we have applied the Coalitional SVM Game on two sets of synthetic data as well as a real world dataset. For our experiments using synthetic data we started with initial Gaussian data distributions comprised of two classes of separable data, each consisting of 25 data points. We first ran the Coalitional SVM Game on this initial data in an online training manner randomly selecting the player-player-reference triple each iteration of game play. Next, to simulate virtual concept drift, we added an additional 25 data points to each class shifting the means of the data distributions and resumed game play using the now larger datasets. Fig. 4 depicts the results of the dynamically changing concept drift data rotating a linear discriminant. The top half of the figure illustrates the initial data classes as well as the resulting discriminant for the initial data. The lower half of the figure illustrates the expanded data classes where each distribution has drifted laterally. The original discriminant is shown as a light gray dashed line with the updated discriminant depicted as the horizontal black line shown.

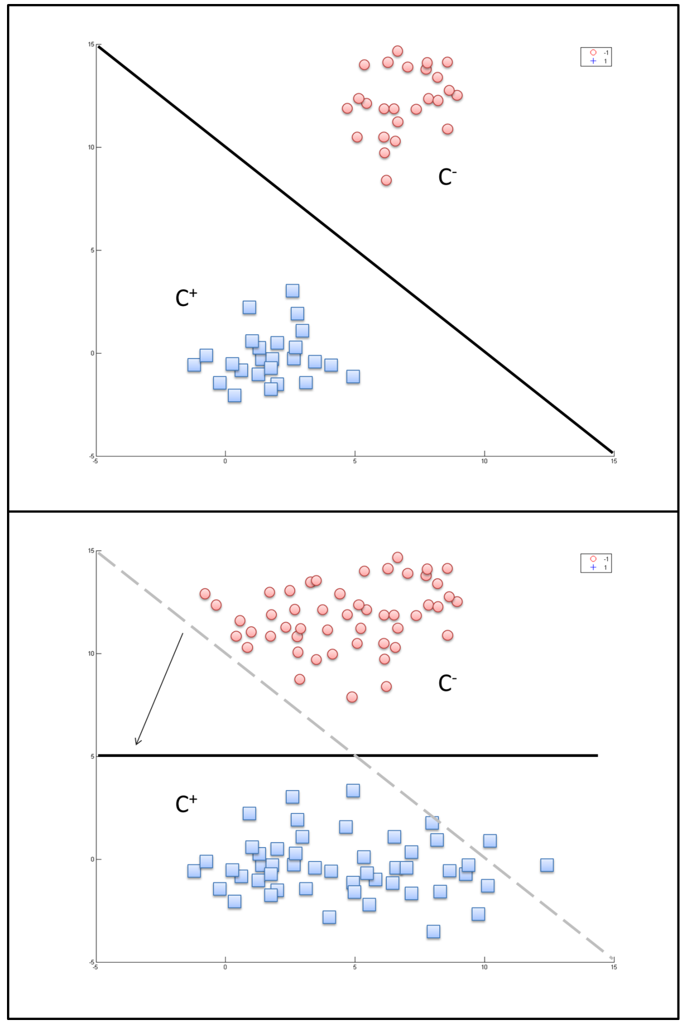

Fig. 4. Dynamically Changing Data Rotating a Linear Discriminant

As a second example of a more challenging virtual concept drift scenario, we once again started with 2 Gaussian separable distributions of 25 data points per class and ran the Coalitional SVM Game. As we updated the distributions to simulate concept drift, while one of the classes simply expanded about its mean, we made the other distribution bimodal. Fig. 5 illustrates the original linearly separable data in the top half of the figure as well as the updated data in the bottom half. As shown, the original discriminant is not simply rotated as before, but rather is updated to become a piecewise linear discriminant.

In addition to synthetic data, we have also applied the SVM Game algorithm to real world data showing concept drift over imbalanced sets with the drift occurring across a temporal progression. The Auto MPG Data Set provides automotive statistics for 398 vehicles from 1970 to 1982 [19]. From this dataset we extracted the four and eight cylinder vehicles as our data classes. For illustrative purposes we also reduced the dimensionality of the problem to two, focusing upon the vehicle weight and fuel economy as our features. Partitioning the data temporally allows us to use this data in a concept drift setting such that over time technological advances have led to increased fuel economy and changes to vehicle weight. We have divided the data into three temporal segments: 1970-1974, 1970-1979, and 1970-1982 where each partition extends the prior partition with the addition of the data from the ensuing years. Fig. 6 depicts the three temporal segments as well as the resulting non-linear discriminants.

V. Discussion

In the results we have shown here, operating the SVM Game in an online learning paradigm through repeated game play allowed the algorithm to update the classification discriminant without needing to discard the current solution and start anew with the addition of the concept drift data. This is not necessarily the case for some learning algorithms. In its canonical form, the fundamental SVM algorithm would need to re-run its quadratic optimization problem on the expanded dataset. Our SVM Game algorithm however, is able to update and continue learning without discarding prior information. Some learning algorithms, such as backpropagation, may lose formerly learned memories as they are updated to incorporate new data. This ability to continue learning while retaining existing information is known as the stability-plasticity dilemma. Since

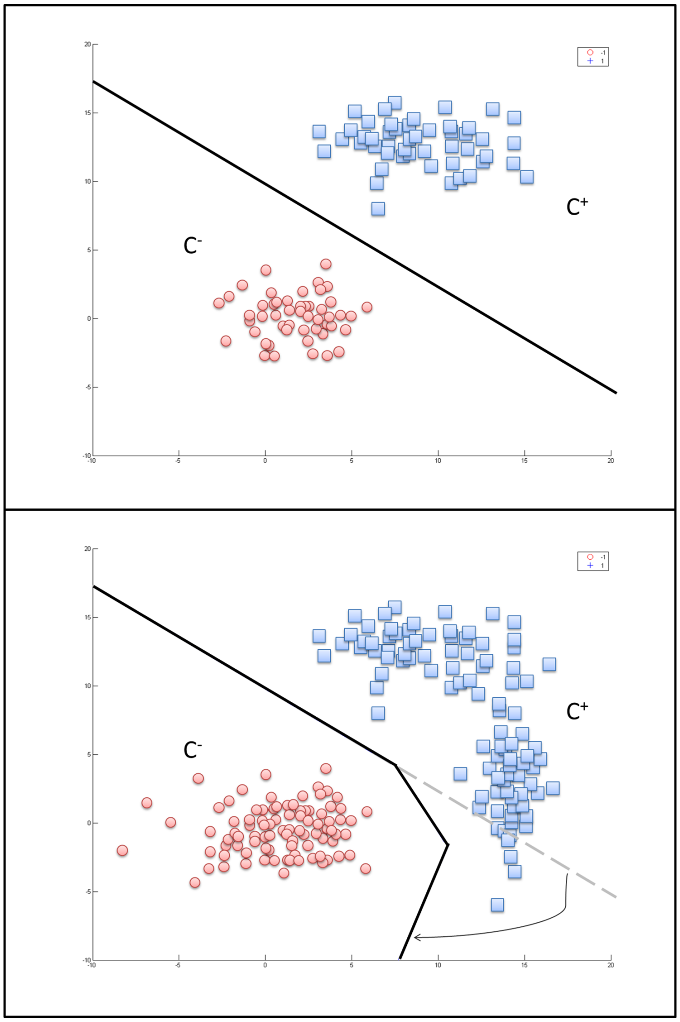

Fig. 5. Dynamically Changing Data Curving a Discriminant

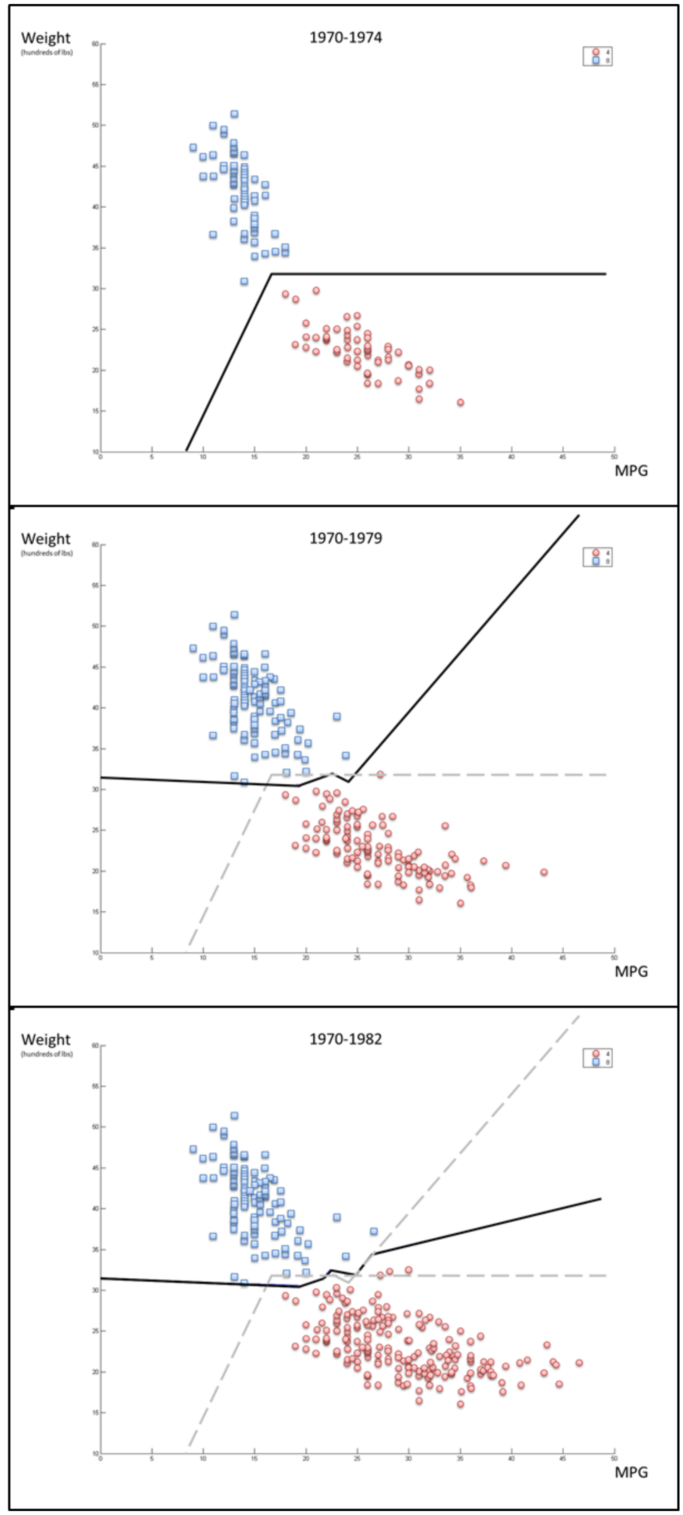

Fig. 6. Real World Automobile Data Example

all individual game iterations are independent, the inclusion of new data points does not invalidate the results of existing games. Depending upon the placement of the new data points, the preference orderings of players may change, and in effect by playing future game iterations which involve the added data points the players may update their coalition preferences accordingly.

In a concept drift setting, the transformation of the data dictates the resulting change, if any, to the SVM Game discriminant. If the respective distributions of two classes being differentiated drifted in a manner such that the boundary between them does not change, then the new data may be incorporated into the overall dataset without yielding any new coalitions and effectively causing the discriminant to change. However, when the updated data distributions cause a change to the set of coalitions, this represents a change to the boundary between the classes. As a result, the discriminant yielded by the SVM Game changes accordingly. As our first example illustrated (Fig. 4) the resulting effect simply rotated the linear discriminant. It is also possible that the resulting transformation could reposition a linear discriminant. In either case, a linear discriminant is generated when each class yields a single coalition.

In more complex adaptations, a linear discriminant may be adapted into a non-linear discriminant, or the shape of a non-linear discriminant may change. A non-linear discriminant arises from the presence of more than one coalition in at least one of the classes. As our second example shows (Fig. 5), what was originally a linear solution becomes a piecewise linear curved discriminant as new coalitions are formed to account for the emergence of a bimodal distribution through concept drift. Specifically, in this example, classes C^+ and C^- both expanded from one to three coalitions each. Our third example on real world data (Fig. 6), illustrates a scenario in which multiple coalitions already exist, yielding a non-linear solution, however as the coalitions change likewise so does the non-linear solution. New coalitions may emerge and/or existing coalitions may be subsumed to change the curvature of the non-linear discriminant. In the automotive example, across the temporal segments the number of coaltions for four cylinder and eight cylinder automotive classes respectively was $\{2,1\}$, $\{3,3\}$, and $\{4,5\}$. In transitioning from the first temporal segment to the second, the original two coalitions for the four cylinder class are both subsumed by a new coalition closer to the boundary between the classes, and two new coalitions are added entirely. From the perspective of the eight cylinder class, one of the newly formed coalitions is an already existent data point in the first temporal segment. However, while the topological configuration in the original data distributions did not result in it being a coalition originally, with the changes to the distributions arising from concept drift it became a coalition in the second solution. The additional two coalitions are a newly formed coalition from the new data as well as the continued presence of the original coalition in the first temporal segment.

Not only may concept drift shift, rotate, or transform a discriminant based upon the resulting impact on the underlying coalitions, but additionally the dynamic behavior of the distributions may introduce noise or cause the distributions to overlap. We can address this issue with the SVM Game using preprocessing approaches such as the Kernel trick and Wilson edits. The Kernel trick is a method commonly employed by SVM to address both non-linear and non-separable problems [20]. Developed by Vapnik et al., the Kernel trick casts the data from input space to a higher-dimensional feature space [10]. The goal of this dimensionality increase is to transform the data, using an appropriate kernel, to a domain such that the data is linearly separable in the higher-dimensional space and can be addressed using the standard SVM algorithm. Importantly, kernel functions are posed such that the dimensionality of the problem itself is not increased (effectively increasing the computational complexity as well as incurring the curse of dimensionality) but rather a distance metric is computed (via dot products) in the higher dimensional space. Consequently, this same approach may be employed in the SVM Game to compute distances between data points as a fundamental operation of the game play.

The Wilson edits method operates upon labelled data and uses an unsupervised clustering method such as k-Nearest-Neighbor (kNN) or k-Means to identify irregularities in the data [21]. By running the desired clustering algorithm on the data distribution a classification is generated for each data point. The points whose labelled classification contradicts the unsupervised clustering classification are identified as extraneous (whether they are noise, overlapping points, inconsistent data, or simply misclassified) and are removed from the data set for subsequent processing by the SVM Game. By applying techniques such as these, the SVM Game may then address noisy and overlapping data introduced through concept drift. For example, in the real world automotive data, the second temporal segment introduced a few overlapping points between the four and eight cylinder classes. By applying the Wilson edits approach using k-Nearest-Neighbor these overlapping boundary points were removed and the coalitional SVM Game was run on the subsequent separable data.

The examples we have shown in this paper are only in two dimensions for visualization sake. The SVM Game algorithm however simply relies upon a distance function and may be applied to higher dimensional problems as desired.

VI. Conclusion & Future Work

In game theory, repeated games provide a means of assessing long term interactions. By extending game play to include potential future interactions players must take into account more than just the immediate consequence of their chosen actions. The question thus arises, at what point should a player change their strategy? In many cases this is driven by trigger effects, threats, and reciprocity [8]. In the context of the concept drift paradigm one must likewise address at what point should a classifier or model be updated. Much like a trigger effect may cause a player to switch their strategy in response to the play of an opponent, one technique is to use an existing classifier until an acceptable error tolerance is exceeded at which point the classifier must be updated.

Or alternatively, rather than basing the trigger upon classification accuracy, one may also view the addition of data points (unbeknownst of whether they further consolidate or alter the existing data distributions) as a trigger. It is this later approach we have presented in this paper although the first approach could be employed with the SVM Game as well. In all of our results shown here, having received the first set of data, the coalitional SVM Game is able to run until the coalitions stabilize as a heuristic based stopping criteria. Upon receiving the subsequent data, the algorithm resumes and once again is run until the coalitions stabilize to yield the updated discriminant. Although the coalition partners learned in a prior phase of data are retained in the subsequent phases, for these examples, we have re-initialized all alpha when adding subsequent data sets. Alternatively, the existing alpha values could be re-normalized to allow alpha to be distributed to the new points while maintaining the relative alpha scaling already learned.

In the work we have presented here, all data points are equally likely to be selected as players in a given game iteration. But while triggering effects in game theory result in changes in strategy, it is not always the case that the strategy change is indefinite. Rather, a temporal window may be employed upon which the new strategy is applied. Likewise, rather than including all data points in the SVM Game whenever an update is triggered, it may be possible to employ various approaches such that particular data points are more likely to be selected based upon factors such as a temporal component or their significance. The canonical SVM algorithm fundamentally only relies upon a subset of data (the namesake support vectors). These influential data points are not known ahead of time or the solution would not need to be computed. In the work by Klinkenberg, he applied the canonical SVM algorithm to concept drift by using an adaptive time window on training data and by weighting training example importance to cut down the computational cost of rerunning the SVM algorithm as the data drifts [22]. As future work, we can apply similar approaches to the selection of players involved in a game iteration as well as the retention of coalitions. For example, just as Klinkenberg utilizes an adaptive time window, the temporal recency of data points may influence their selection in an iteration of the SVM Game. And similarly, if only the most temporally recent data points are included in the SVM Game, likewise a recency component may be incorporated to drop old coalitions if they do not pertain to the recent data. Furthermore, Klinkenberg also employs example weighting approaches in which instead of simply using a temporal weighting, data is evaluated based upon other heuristics such as how representative they are of the class. Applying such techniques in the SVM Game could be based upon factors such as the amount of alpha players have or the relative strength of their coalition. Ultimately, applying these or related techniques would complement the Klinkenberg work by applying their techniques to the SVM Game as an alternative to the canonical SVM algorithm. Doing so has potential benefits such as the distributed nature the SVM Game confers as well as its innate approach to concept drift through repeated game play. Approaches such as these which constrain the availability of data are examples of stream classification algorithms.

Additionally, the SVM Game could also be utilized as a base classifier within general concept drift frameworks such as the Learn^{++}.NSE ensemble approach and the Just-in-time (JIT) classifier [23][24]. The Learn^{++}.NSE framework is a batch learning ensemble approach that is incrementally trained. As data is received, a new classifier is trained and incorporated within the ensemble. The Learn^{++}.NSE framework is not a classifier itself, but allows for a desired base classifier to be used. As such, the SVM Game may be utilized as the base classifier within the Learn^{++}.NSE framework. Likewise, the JIT classifier is a general framework which in the absence of detected concept drift (i.e. the world is stationary) utilizes newly received information to improve upon its current classification accuracy. Upon detecting concept drift through change-detection tests (CDTs), a new concept representation is created and incorporated within the overall classifier. Using the SVM Game as the underlying base classifier confers the possibility of relating the CDT with the intrinsic operation of the SVM Game itself, such as based upon coalition related trends. And furthermore, the SVM Game would provide additional techniques for updating, splitting, or merging concepts based upon the SVM Game-play directly.

Besides experimenting with various means of constraining data availability and using the SVM Game as the base classifier for general concept drift frameworks, we would also like to assess repeated play of the SVM Game on higher dimensional data as well as standard concept drift datasets such as SEA and RBF Generator with respect to synthetic datasets, and the Email Spam real-world dataset [3]. The Random RBF Generator dataset may be particularly interesting as Scholkopf et al. have shown how the canonical SVM algorithm may be used to train a Radial Basis Function (RBF) network by relating support vectors to RBF centroids [25]. Consequently, as an iterative and adaptive approach to SVM, the SVM Game may be quite amenable to this dataset.

In essence, the ability for the SVM Game to operate upon dynamic data is evident by the proof of convergence to the geometric SVM linear solution when a unanimous Condorcet winner exists given in [18]. This inductive proof iterates over data set size, starting with two points in a data class as the base case and showing the algorithm properly addresses all possible placements of further data points. Just as the inductive proof is based upon the premise of correctly handling the addition of new data points in an agglomerative manner, given sufficient game iterations, a non-stationary dynamic data set is simply an extension of this notion. Since the SVM Game iterations are independent of one another, while the SVM Game has not yet converged, data points which have not been included in any game iterations, but are known a priori to the algorithm are fundamentally equivalent to new data. The use of the mathematics of game theory and mechanism design in relation to machine learning problems such as classification provides descriptive insight as well as functional benefit. Game-theoretic algorithms are often iterative by nature, consisting of repeated game play rather than a single interaction. Effectively, rather than requiring extensive retraining to update a learning model, a game-theoretic approach can adjust strategies as a novel approach to the concept drift problem. As presented here, by operating in an online learning and online training paradigm the SVM Game algorithm through repeated game play is well suited for adaptive classification such as virtual concept drift.

ACKNOWLEDGMENT

This research was possible in part by LDRD program support from Sandia National Laboratories. Sandia National Laboratories is a multi-program laboratory managed and operated by Sandia Corporation, a wholly owned subsidiary of Lockheed Martin Corporation, for the U.S. Department of Energys National Nuclear Security Administration under contract DE-AC04-94AL85000.

REFERENCES

[1] A. L. Samuel, “Some studies in machine learning using the game of checkers,” *IBM Journal of research and development*, vol. 44, no. 1.2, pp. 206–226, 2000.

[2] J. C. Schlimmer and R. H. Granger Jr, “Incremental learning from noisy data,” *Machine learning*, vol. 1, no. 3, pp. 317–354, 1986.
[3] J. Gama, I. Žliobaitė, A. Bifet, M. Pechenizkiy, and A. Bouchachia, “A survey on concept drift adaptation,” *ACM Computing Surveys (CSUR)*, vol. 46, no. 4, p. 44, 2014.
[4] M. Salganicoff, “Tolerating concept and sampling shift in lazy learning using prediction error context switching,” *Artificial Intelligence Review*, vol. 11, no. 1-5, pp. 133–155, 1997.
[5] X. Zhu, “Semi-supervised learning literature survey,” 2005.
[6] A. Tsymbal, “The problem of concept drift: definitions and related work,” *Computer Science Department, Trinity College Dublin*, 2004.
[7] L. J. Neumann and O. Morgenstern, *Theory of games and economic behavior*. Princeton university press Princeton, NJ, 1947, vol. 60.
[8] P. K. Dutta, *Strategies and games: theory and practice*. MIT press, 1999.
[9] M. Osborne and A. Rubinstein, *A Course in Game Theory*. MIT Press, 1994.
[10] V. N. Vapnik and V. Vapnik, *Statistical learning theory*. Wiley New York, 1998, vol. 2.
[11] R. O. Duda, P. E. Hart, and D. G. Stork, *Pattern Classification (2nd Edition)*, 2nd ed. Wiley-Interscience, 2001.
[12] K. P. Bennett and E. J. Bredensteiner, “Duality and geometry in svm classifiers,” in *Proceedings of the Seventeenth International Conference on Machine Learning*, ser. ICML ’00. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 2000, pp. 57–64. [Online]. Available: http://dl.acm.org/citation.cfm?id=645529.657972
[13] E. T. Noam Nisan, Tim Roughgarden and V. V. Vazirani. Cambridge University Press, 2007, ch. 9.
[14] J. Kleinberg, *Networks, Crowds, and Markets*. Cambridge University Press, 2010.
[15] C. M. Vineyard, G. L. Heileman, S. J. Verzi, and R. Jordan, “Game theoretic mechanism design applied to machine learning classification,” in *Cognitive Information Processing (CIP), 2012 3rd International Workshop on*. IEEE, 2012, pp. 1–5.
[16] K. Mitchell and J. Ryan, “Game theory models of animal behavior.”
[17] J. H. Smith, “Aggregation of preferences with variable electorate,” *Econometrica*, vol. 41, no. 6, pp. 1027–41, 1973.
[18] C. M. Vineyard, G. L. Heileman, and S. J. Verzi, “A Game Theoretic Coalitional Support Vector Machine Classifier,” 2015, under Review.
[19] K. Bache and M. Lichman, “UCI machine learning repository,” 2013. [Online]. Available: http://archive.ics.uci.edu/ml
[20] S. Haykin, *Neural Networks: A Comprehensive Foundation*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 1998.
[21] D. L. Wilson, “Asymptotic properties of nearest neighbor rules using edited data,” *Systems, Man and Cybernetics, IEEE Transactions on*, no. 3, pp. 408–421, 1972.
[22] R. Klinkenberg, “Learning drifting concepts: Example selection vs. example weighting,” *Intelligent Data Analysis*, vol. 8, no. 3, pp. 281–300, 2004.
[23] R. Elwell and R. Polikar, “Incremental learning of concept drift in nonstationary environments,” *Neural Networks, IEEE Transactions on*, vol. 22, no. 10, pp. 1517–1531, 2011.
[24] C. Alippi, G. Boracchi, and M. Roveri, “Just-in-time classifiers for recurrent concepts.” *IEEE transactions on neural networks and learning systems*, vol. 24, no. 4, pp. 620–634, 2013.
[25] B. Scholkopf, K.-K. Sung, C. J. Burges, F. Girosi, P. Niyogi, T. Poggio, and V. Vapnik, “Comparing support vector machines with gaussian kernels to radial basis function classifiers,” *Signal Processing, IEEE Transactions on*, vol. 45, no. 11, pp. 2758–2765, 1997.