SVM integrated Case Based Restarting GA for further Improving Solar Flare Prediction

Yukiko Yamamoto, Daichi Itoh, Setsuo Tsuruta
School of Information Environment
Tokyo Denki University
Inzai, Japan
tsuruta@mail.dendai.ac.jp

Yuko Hada-Muranushi
Kwasan and Hida Observatories
Kyoto University
Kyoto, Japan
yukorin@kwasan.kyoto-u.ac.jp

Yoshiyuki Mizuno
Faculty for the Study of Contemporary Society
Kyoto Women's University
Kyoto, Japan
e-mail: mizuno@kyoto-wu.ac.jp

Takayuki Muranushi
RIKEN Advanced Institute for Computational Science
Kobe, Japan
takayuki.muranushi@riken.jp

Syoji Kobashi
Graduate School of Engineering
University of Hyogo
Himeji, Japan
kobashi@eng.u-hyogo.ac.jp

Rainer Knauf
Faculty of Computer Science and Automation
Ilmenau University of Technology
Ilmenau, Germany
rainer.knauf@tu-ilmenau.de

Abstract—Solar activity has various influences on the global environment. Specifically, it may have serious impacts on the Earth such as satellite damage, etc. and power plant failures causing more serious disaster. For a precise forecast of larger scale solar flares causing serious disaster, it is important to improve the space weather forecast, a daily forecast of the solar flare. In our work so far, a machine-learning algorithm called Support Vector Machine (SVM) was used. We extended this technology by integrating Case Based Genetic Algorithm (CBGA) for a more precise forecast. It was shown experimentally that triple mutation rate on the slowdown of evolution in our CBGA improves considerably (e.g. another 5%) more than original mutation rate in the True Skill Statistics TSS. For further obtaining the optimality towards more imbalanced data analysis applicable to the recognition of serious disaster or medical disease, Restart CBGA is proposed with its expected effect. Here GA integrating SVM is restarted using highly optimized but diversified solutions in the case base as initial individuals. Further this restart CBGA is repetitively and evolutionary performed, evolving and maintaining the case base by the result of each (restarted) GA.

Keywords-solar flare forecast, imbalanced data, Support Vector Machine, Genetic Algorithms, Case Base, Restart

I. INTRODUCTION

The prediction of the onset of solar flares [22] is a big challenge in current space weather studies [4][21]. Such studies are necessary, because unusual solar flares may cause serious impacts to the Earth such as failure of satellite communication and navigation (GPS), satellite damage, increased radiation exposure to astronauts, geomagnetic storm and aurora, and power plant failures causing more serious disaster, see Fig. 1.

Fig. 1. The need of space weather forecast

There are several works on flare prediction based on dynamical models [18][22], but the state of the art is far away from a reliable and precise prediction [5]. Alternatively, empirical algorithms based on flare onset in the past could be used to calculate the probability of flare occurrence [20], for example by a discriminant analysis of magnetic parameters from vector magnetic fields [14][15], superposed epoch analysis [18], and Bayesian statistics [26].

Recently, Knowledge Engineering Techniques such as Support Vector Machines (SVM) [16][7][9], Logistic Regression [24][28], optimized regression of time series [20], and Neuronal Networks [11][27][1] have been proposed to solve this problem. Flare forecasting work so
far use either line-of-sight magnetograms or a small number of ground-based vector magnetograms.

SVM [9] is a binary classification learning technique. A catalog of flaring and non-flaring active regions sampled from a database of 2,071 active regions, composed of 1.5 million active region patches of vector magnetic field data, could be established. And active regions could be characterized by 25 parameters.

Typically, the two classes are called positive and negative. By convention, the positive class denotes flaring active regions, and the negative class denotes non-flaring active regions. To train and test the SVM, the catalog of positive and negative examples is separated into two, non-overlapping, datasets. Li et al. [16] and Yuan et al. [28] used a soft margin SVM algorithm to forecast solar flares.

Bobra et al. [7] use Scikit-Learn module implementation of a soft margin SVM in the Python programming language. In solar-flare forecasting, the two classes (non-flaring and flaring regions) are strongly imbalanced: there are many more negative examples than positive ones. This class imbalance is a major issue for most machine-learning algorithms. Indeed, a ML classifier may strongly favor the majority class, and neglect the minority one. In other words, always predicting that a Crucial this issue

Several solutions are presented in the literature to remedy this issue [17]. The easiest way is to under-sample the majority class, i.e. to build a training set that has about as many negative as positive examples. However, this leads to poor results, since non-flaring regions can exhibit quite a large range of features that are not captured by a small sample. Moreover, the flare catalog developed in [7] is relatively small and requiring a perfect balance between classes would lead to only using a small number of non-flaring regions.

A better way is to assign different cost parameters to the two classes. This has been done in [7] in and is a functionality offered by the SVM as a class of Scikit-Learn. This implementation is both robust and fast. The SVM can be either linear or non-linear. A linear classifier seeks to separate the examples by finding a hyperplane in the feature space.

To use non-linear decision functions to separate the examples, i.e. not a hyperplane but a more complicated hyper-surface, [7] use kernels that remap the feature space into a higher-dimensional space. Whether in the original feature space or in the remapped one, the SVM tries to find the separating hyperplane with the largest distance to the nearest training examples: it is a maximum margin classifier.

The SVM was trained and tested and its performance was estimated using forecast verification metrics called “True Skill Statistic (TSS)” [7] because it is adopted as a standard [6] for forecast comparison over conventionally used Heideke skill score (HSS).

Here, for more precise forecasting, we extended this SVM application technology as follows:
1) integrated with Genetic Algorithm (GA) enhanced by Case Based Technology
2) mutation rate of the GA being dynamically changed.

The effect of this approach is evaluated experimentally using TSS mentioned above.

As far as experimentally validated, the combination of 3 population and 3 times of mutation rate (3%) in case of the cease or stagflation of evolution obtained the considerably good optimality as shown concretely in the later section. However, it is attained at the 3rd or early generation and then falls into local optimum. The tendency is mostly the same as shown later in figure 7-9.

This suggested us the case based diversity approach GA here we call Restarting CBGA or CBGAO for further obtaining the optimality towards more imbalanced data analysis applicable to the recognition of serious disaster or medical disease. Namely, in case the evolution ceased more than 3 generations or especially the elite equivalent to the top level individual in case base is obtained, the GA should stop. Then the next GA process has to start (restart) retrieving and using the highly optimal or top level but diversified (different each other) individuals in the case base as initial individuals. Then such stoppage and restart of GA should be performed for more efficient optimization with less stagflation.

Thus GA integrating SVM is restarted using highly optimized but diversified solutions in the case base as initial individuals. Further this restart CBGA is repetitively and evolutionary performed, evolving and maintaining the case base by the result of each (restarted) GA.

The paper is organized as follows. The section II sketches the technology used so far to forecast the space weather. In particular, the data input and output is described. Section III introduces the concept of Genetic Algorithms and its use to improve this technology by supporting the (so far human made) “guesses” to compose so called image patterns towards a complete forecast. In section IV, we describe the Case-Based technology in general and its use for this application. Section V describes the experimentally evaluated effects of our optimal method to dynamically change mutation rate, its limitation and the new approach (Restarting CBGA or CBGAO) to cope with it. Section VI concludes the paper.

II. DATA PREPARATION, PROCESSING, AND INTERPRETATION OF THE SVM APPROACH SO FAR

The solar flare pictures are taken frequently over years and the objective of its analysis is to predict the flares in the future based on the data of the past. The time unit between two pictures is usually one hour, which is sufficiently smaller than the evolution timescale of individual sunspot [20].
Such pictures are available on databases such as a one of the Joint Science Operation Center (JSOC) [10] with a resolution of 4096 x 4096. To suppress the noise, pixels with a large distance from the center are removed. After that, the picture is scaled down to 1024 x 1024 [20].

To become subject to human expert interpretation and discussion as well as to become subject to machine knowledge processing, for these pictures, the complexity of solar features needs to be quantified.

Solar flare productive sunspots differ from non-flare productive sunspots not just in the radiation intensity per area (measured in \( W/m^2 \)), but also much more rapid changes of the radiation intensity, see Fig. 2.

![Non Flare-productive sunspot: simple, stable](image1)

![Flare-productive sunspot: complex, rapidly evolving](image2)

**Fig. 2. Feature selection by image processing**

The feature vectors are generated by a 2D discrete wavelet transformation [2] that generates to 11 (\( 1024 = 2^{10} \)) distinct regions per edge of the wavelet space. Wavelet transformation is good in time resolution of high frequencies, while for slowly varying functions, the frequency resolution is remarkable. Thus we used a 2D discrete wavelet transformation not related to time frequency.

For the output data, we use GOES soft X-ray flux data available at the Space Weather Prediction Center Website [34]. The X-ray flux is a variable integrated over the entire solar image and shows peaks from particular flares, which allow flare prediction. Since it is quite difficult to predict individual elements, the flares are often treated as stochastic events and thus, we use statistical variables as the target of the prediction.

The prediction itself is mostly a classification of upcoming flares within a certain time horizon. In [20], for example, they predict three different classes of upcoming flares, namely (1) \( X \), which is a positive event, if there will be a future maximum of X-ray flux of more than \( 10^{4} \) \( W/m^2 \), (2) \( \geq M \) which is a positive event, if there will be a future maximum of X-ray flux of more than \( 10^{5} \) \( W/m^2 \), and (3) \( \geq C \) which is a positive event, if there will be a future maximum of X-ray flux of more than \( 10^{6} \) \( W/m^2 \) within the next 24 hours by two regression machines, the SVM library LibSVM [8] and a simple own linear regression algorithm.

III. PROPOSED GENETIC ALGORITHM

Genetic Algorithms (GA) are often used as approximation algorithms to solve NP complete problems nearly optimally with an acceptable degree of computational complexity [3]. Additionally, we propose the combination of the GA with a Case Based Reasoning (CBR) technology for intelligent Knowledge Acquisition.

Here, the basic idea to improve the forecast is to gradually develop better and better solutions, not by manually composing a forecast out of the patterns, but by evolutionary composing them from good solutions in a former generation of potential solutions by a GA, see Fig. 3.

![Integration of GA to improve the forecast](image3)

**Fig. 3. Integration of GA to improve the forecast**

The conceptual idea to improve the local optimization in the highly imbalanced date of this application is as follows:

1. By combining GA (Genetic Algorithm) with CBR (Case Based Reasoning), we evolutionarily compute the better architecture of inherently local machine-learning/optimization methods with their application strategy (e.g. application orders, confliction resolver, or feature/parameter selection patterns).
2. For the initial population of our GA, we use formerly successful cases such as good methods or good strategies including parameter selections from previous creations of potential solutions for similar problems.
3. This is performed in cooperation with expertise or experts, who revise cases for more improvement, often considering even cultural factors.

This idea is realized as a Case Based GA combined with a Local Optimizer (CBGALO). The effect for the highly imbalanced class distribution and/or global optimization is fundamentally certified in one or two applications for both quality metrics precision and recall. In CBGALO, local optimizing with machine learning methods such as SVM is still weak in data with highly imbalanced
class distributions. Therefore, we use the SVM technology just in the evaluation step. Instead, the GA is used to generate a generation of several predictions by combining the patterns, which are the outcome of the wavelet transformation of the solar X-ray picture preprocessed in the manner introduced in section 2. In other words, each picture can be represented as a set of such patterns.

The SVM is integrated into the GA as the evaluation step, which computes not only the solution but also the fitness of individuals representing such methods with their application strategies. Then, we decide which ones should be selected to be a part of the next generation, fundamentally based on the fitness.

Here, a candidate solution is represented as an individual of the GA. Beginning with an initial population of individuals, genetic operators like selection, crossover and mutation are performed in a cyclic way until a population contains a number of acceptable individuals out of which the best one can be output as the final forecast.

The selection operator selects “good” individuals, allowing them to pass on their genes to the next generation. The goodness of an individual mostly depends on its fitness, but other criteria may be considered, too.

The crossover operator takes two individuals that are chosen from the population. The new offspring created from it mating by using patterns of both parents are put into the next generation of the population. By recombining parts of good individuals, this process is to create even better individuals. The mutation operator produces a new individual by the manipulation of one in the current population by replacing some of its patterns.

The proposed GA is sketched in Fig. 4. Our mutation operator creates new individuals by changing parts of particular individuals. The cross over operator creates new individuals from more than one individual.

The fitness evaluation is performed by (1) executing the SVM using the feature (pattern file) combination specified by the gene and (2) cross validation with using half of the population for building the model and the other half for its validation. So the SVM is embedded into the GA and performs a supervised learning and determines a linear class separation line, which maximizes its distance from each point (in the example sketched in figure 5, it is H2).

So the SVM is embedded into the GA and performs a supervised learning and determines a linear class separation line, which maximizes its distance from each point (in the example sketched in Fig. 5, it is H2).

![Fig. 5. Linear classification by the SVM](image)

In our application, an individual consists of a sequence of genes, which are either 0 or 1, where 1 expresses the use of the feature and 0 expresses not to use it, see Fig. 6. Altogether, there are 100 - 1000 features, each corresponds to a file that contains time series of observational data.

![Fig. 6. Method/Strategy/Feature selection vector](image)

Here, we “forecast” an image of an upcoming solar flare picture by composing several image patterns from a pattern repository. Optimizers or machine learning methods with their application orders/strategies and feature selection patterns are coded on genes of each individual as shown in their (methods/strategies/features) binary selection vector as illustrated in Fig. 6.

Here, an individual consists of a sequence of genes whose values are either 0 or 1, where 1 expresses the use of the method/strategy/feature and 0 expresses not to use it. There can be over 1000 genes, for simple example, each corresponding to a file that contains time series of observational image or thermograph data.

Here, an individual consists of a sequence of genes whose values are either 0 or 1, where 1 expresses the use of the method/strategy/feature and 0 expresses not to use it. There can be over 1000 genes, for simple example, each corresponding to a file that contains time series of observational image or thermograph data.
From case bases, good but various candidate solutions whose problems are similar are selected and modified to be represented as initial individuals of the GA [12] whose ways are also concretized later or after the research advances. The initial populations or cases are also generated or modified / revised, based on various combinations of patterns proposed by human experts. This is some kind of knowledge fusion of individual expert’s knowledge toward a set of potential prediction. Thus, the initial generation (initial population) of our Case Based GA can be a set of “good guesses” composed of patterns proposed by experts or expertise and thus, some sort of collective knowledge from many experts.

Beginning with an initial population of individuals, genetic operators like selection, crossover and mutation are performed in an iterated way until a population contains a number of acceptable individuals out of which the best one can be output as the final solution or prediction. Of course, at the end of the last generation, our CBGALO characteristically stores best-level good but various individuals namely solutions set with problems as cases [12], whose way is also described in the next section.

IV. A CASE BASED GA FOR CBGALO

CBR (Case Based Reasoning) is a usual problem solving method in fields, where creating a new solution to a problem from scratch is expensive especially in imbalanced classification problems and solutions to similar problems are available: justice, medicine, architecture, scheduling, data analysis, and even bigger programming projects and others. This is performed by holding a CB (Case Base) with pairs [problem, solution] of formerly solved problems and consists of the steps (1) case retrieval, (2) case reuse, (3) case revision, and (4) case retaining. Using CBR for GAs was formerly proposed in [12] and these four steps were defined as follows.

A. Case Retrieval

If a new or actual problem has to be solved, a “most similar” problem needs to be retrieved from the CB. For such retrieval, similarity of a case (solution) ti to the given new problem p is defined as sim(ti, p).

In the application field of the work in this paper, most of the elements in the binary feature vector are 0. Therefore, a proper similarity metrics should consider only elements, in which at least one of the considered individuals holds the value 1. For this reason the Jaccard-Coefficient

\[
J = \frac{f_{11}}{f_{01} + f_{10} + f_{11}} \quad (1)
\]

should be used as the similarity metrics between two Method / Strategy / Feature selection vectors as illustrated in figure 6. If there are several cases each having the same degree of similarity as a problem but several solutions, a case with the best fitness is retrieved as the “most similar” case. However, a less similar solution may also be appropriate too for maintaining diversity, especially at least combined with the consideration of its fitness f. Therefore, we retrieve the case with the next lower degree of similarity (respective the fittest one among them, if there are several ones) and look, whether this case has a higher fitness than the most similar case.

We define fitness gain of a solution with higher fitness \(f_i\) related to a solution with lower fitness \(f_s\) as \((f_i - f_s) / f_i\). Also, we consider the similarity loss of a solution (case) to the retrieved “most similar” solution (case) \(s\), related to the (next similar) solution \(s_2\) with the next lower degree of similarity \(\text{sim}_2\) to “most similar” \(s\) as \(\text{sim}(s, s_2)\).

The next similar solutions are searched until coming to a point, at which the next similar solution has a higher similarity loss than fitness gain. On reaching this point, the next similar solution is refused and the lastly considered solution is retrieved.

B. Case Reuse and Revision

After case retrieval, the case is reused but in case of necessity some automatic / (in this proposal just) manual revision can be done for faster or more improvement by GA operations.

C. Case Retaining and Case Base Maintenance

Case retaining or inclusion (case base maintenance) issue is solved as follows. First we determine the most similar case in the Case Base so far.

1. If \(f_{\text{new}} \geq f_i\) and \(\text{sim}(s, s_{\text{new}}) < 1\), the new solution is accepted for inclusion into the CB (\(f_s\) is for candidate case).

2. If the solution has been modified by humans to meet side conditions including human and cultural issues, it is also included into the CB irrespective of its fitness.

3. Otherwise, the new solution is not included.

If the CB size exceeds the above discussed reasonable value of \(n\) being the largest scale included in the CB and solved by the system, the lowest fitness solution is removed from the CB.

V. EVALUATION AND IMPROVEMENT

We followed the suggestion of Bloomfield et al. [6] as well as Bora and Couvidat [7] to evaluate the performance of the flare prediction by the True Skill Statistics TSS. This measure is based on the contingency table (see table 1), which contains the numbers true positive (TP), false positive (FP), true negative (TN), and false negative (FN) predictions.

<table>
<thead>
<tr>
<th>TABLE I. CONTINGENCY TABLE</th>
</tr>
</thead>
<tbody>
<tr>
<td>event</td>
</tr>
<tr>
<td>predicted “yes”</td>
</tr>
<tr>
<td>predicted “no”</td>
</tr>
</tbody>
</table>

Based on this table, the True Skill Statistics TSS is calculated by

\[
\text{TSS} = \frac{TP}{TP + FN} - \frac{FP}{FP + TN} \quad (2)
\]
There are two reasons that make it reasonable to quantify the performance of the prediction by this metrics, namely:

1. TSS is one of the metrics that equals to 0 in case the predictor has no knowledge about the event and equals to 1, if the prediction is perfect, i.e. if FP = FN = 0.
2. TSS is one of the metrics, which is not affected by the ratio between the numbers of positive and negative event. It has no problem with class imbalances, which typically occur in solar flares, because most of the time the space weather doesn’t cause any problems on earth.

Because of these two reasons, this metrics has been used as an a good indicator to compare predictors in previous studies, too [6][7][20].

In a first experiment, we considered the improvement with a fixed population size over 25 generations. To face the problem of falling into local optima, we multiplied the mutation rate, whenever there was no significant progress in the TSS from one generation to the next.

Fig. 7 shows the TSS development, when the mutation rate was doubled after the 2nd generation. As a result, the TSS went up 12.8% from 0.484 to 0.546.

![Fig. 7. TSS development with mutation rate doubled](image)

In Fig. 8, the mutation rate was tripled which resulted in a TSS improvement 18.6% from 0.490 to 0.581.

![Fig. 8. TSS value improvement with mutation rate tripled](image)

Interestingly, giant change of the mutation rate ends up with similar final results with falling into another local optimum as can be seen in Fig. 9 (TSS improvement 17.8% from 0.443 to 0.522).

![Fig. 9. TSS development with mutation rate multiplied by 4](image)

Thus as far as experimentally validated above, the combination of 3 population and 3 times of mutation rate in case of the cease or stagnation of evolution seems the best to obtain the best optimality (TSS value 0.581, 41.7% increase compared with the TSS value 0.410 of initial individuals). However, it is attained at the 2nd generation and falls into local optimum. The tendency is mostly the same as shown also at Fig. 9.

This suggested us the case based diversity approached GA here we call Restarting CBGA or CBGALO for further obtaining the optimality towards more imbalanced data analysis applicable to the recognition of serious disaster or medical disease. Namely, in case the evolution ceased more than 2-3 generations or especially the elite equivalent to the top level individual in case base is obtained the GA should be stopped. Then another GS have to restart using top level but diversified individuals in the case base as initial individuals. Then such stoppage and restart of GA should be performed for more efficient optimization with less stagnation.

Thus GA integrating SVM is restarted using highly optimized but diversified solutions in the case base as initial individuals. Further this restart CBGA is repetitively and evolutionary performed, evolving and maintaining the case base (e.g. in c of Section IV) by the result of each (restarted) GA.

More concretely, in case of CBGALO that is heavy GA including a machine learning mechanism such as SVM, the efficient improvement rules are as follows:

1) The more giant is the improvement, the more shortly should the restarting occur after the stop or stagnation of evolution (continuation of the same fitness value). This is because new individuals cannot easily exceed such a greatly improved elite and the evolution becomes too inefficient.

2) The less is the population size, the more shortly should the restarting occur after the stop or stagnation of evolution. This is because the survival possibility of individual different from such a greatly improved elite becomes significantly low (same individual types as greatly improved elite becomes pervasive) and the evolution becomes very inefficient.
3) Restarting should start at the maximum of 3 generations or the population size after the stop of evolution from our experience as is shown in figure 7-9.

The expected effect of this approach here we call Restarting CBGA as follows:

1) Owing to restart, local optimization in GA namely evolution stagnation can be somewhat easily avoided. Namely, the diversity can be more easily maintained in harmony with fitness, using Case Base as follows:

1-1) Different from a tabu list, individuals (or solutions) already created and evaluated not only in one GA but also in other tries of GA such as restarted GAs can be stored in the case base. Then, solutions (cases) with high fitness as well as diversity are selected as initial individuals from the case base on restarting GA to escape from the local optimum or stopping or stagnation of GA’s evolution. This balanced selection having high fitness and diversity due to Case Base is expected to cause efficient optimality in Restart CBGA.

Meanwhile, in conventional GA having neither Restart mechanism nor case base, usually a group of individuals almost the same as an elite or some elites become pervasive even if having a tabu list. Thus stagnation (cease of evolution) occurs.

Island or distributed model GA can have different type of elites independently. However, it has a tabu list in each island. Due to the distributed model and independency in each island, inherently, it does not exchange the tabu list and the same individuals can be evaluated. Further, elites or a group of elite can imigrate and become pervasive in many island which can stop evolution. Therefore this also can not be so efficient in optimal search.

1-2) Furthermore, not exactly the same but similar problem’s solutions can be reused as initial individuals of GA after adjusting for the new problem. Therefore, various kind of good solutions (good solutions with diversity) can be used for initial individuals on starting or restarting GA. It is expected this can create further highly optimal solutions.

2) Different from random restart GA, optimization is more efficiently attained owing to the use of Case Base’s knowledge or good cases (solutions with high fitness) as initial individuals when restarting GA.

3) On restarting GA, besides automatic retrieval of cases for initial individuals, human experts check, control, and even add their expertise or knowledge as initial individuals or (good) cases. Therefore, more optimal solutions can be searched more efficiently.

VI. CONCLUSION AND OUTLOOK

We extended a Support Vector Machine (SVM) by integrating Case Based Genetic Algorithm (CBGA) for a more precise space weather forecast. It was shown experimentally that triple mutation rate on the slowdown of evolution in our CBGA improves considerably (e.g. another 5%) more than original mutation rate in the True Skill Statistics TSS. For further obtaining the optimality towards more imbalanced data analysis applicable to the recognition of serious disaster or medical disease, Restart CBGA was proposed with its expected effect.

ACKNOWLEDGMENT

This work was supported by JSPS KAKENHI Grant Number 15K00349.

REFERENCES


