Our Heritage

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

Adaptive Cost-Sensitive Online Gradient Descent (ACOG) with Swallow Swarm Optimization (SSO) for Online Classification

N.Kamalraj

Dr.SNS Rajalakshmi College of Arts and Science, TN, India

ABSTRACT: *The cost-sensitive classification and online learning have been well considered individually in data mining and classification, there was very few wide-ranging learning of cost-sensitive online classification in review work. On the other hand, recent traditional algorithms simply focused first-order data of data stream. It is inadequate in tradition, because numerous existing methods have proved with the purpose of integrating second-order data improves the classification results of classifiers. To manage this problem, Adaptive Cost-Sensitive Online Gradient Descent (ACOG) classifier by adaptive regularization is developed recently. On the other hand in ACOG classifier, optimization of the cost function becomes extremely hard task. To handle this problem, Swallow Swarm Optimization (SSO) algorithm is introduced which optimizes the parameters of the cost for online gradient Descent classifier. Reduced error classification results parameters designed for positive and negative samples are optimized by SSO algorithm. SSO algorithm includes of three major types of particles: explorer particles, aimless particles, and leader particles. Every particle has an individual characteristic designed for optimization of the cost parameters designed for inner colony of flying. Every particle shows an intelligent behavior and, continuously, discovers its surroundings by means of a reduced error value. Subsequently designed for improved trade off among the results and effectiveness, additional develop the sketching algorithm, which considerably speed up the computation time by means of moderately small results loss. Hypothetically examine the proposed classifiers and existing algorithms in wide experiments by means of the german and covertype dataset. Classifiers are experimented in MATLAB environment and measured by means of sensitivity; specificity, sum, and computation time.*

INDEX TERMS: Cost-Sensitive Classification, Online Learning, Adaptive Regularization, Sketching Learning, Sparse Representation, Optimization, Swallow Swarm Optimization (SSO).

1. INTRODUCTION

Online learning speaks to a group of proficient and versatile machine learning strategies, which have been widely, examined in machine learning and information mining recently [1-3]. When all is said in done, the objective of this learning is to gradually get familiar with some expectation models to make right forecasts on a stream of instances that show up successively [4]. This learning is worthwhile for its high proficiency and adaptability for huge scale applications, and has been applied to handle online order undertakings in an assortment of true information mining applications.

Accordingly, these strategies are hard to deal with various issues, where datasets are consistently class-imbalanced, i.e., the mistake expenses of tests are essentially extraordinary [5]. To take care of this issue, author have proposed utilizing increasingly important parameters, for example, the weighted sum of recall and specificity [6-7], and the weighted misclassification cost [8] to succeed old ones. Depending on, many batch classification methods are proposed to straightforwardly enhance forecast results for cost- sensitive order over the previous decades [8]. Be that as it may, these batch methods regularly experience the ill effects of poor results and scalability for huge-scale tasks, which make them wrong for online characterization applications.

As results, the Cost-Sensitive Online Classification system [9-10] was earlier introduced towards fill the hole between Online learning and costsensitive Classification. As indicated by this structure, a class of calculations named as Cost-Sensitive Online Gradient Descend (COG) was introduced to straightforwardly update predefined cost- sensitive measurements in view of online gradient descent procedure. Be that as it may, in spite of the fact that COG can deal with the Costsensitive algorithms, it just takes the primary request data of tests. It is clearly inadequate, since numerous ongoing investigations [11] have indicated that far reaching thought with second-request data (i.e., the connections between's attributes) essentially improves the results of classification.

In order to increase the COG algorithm, Adaptive Regularized Cost- Sensitive Online Gradient Descent algorithms (ACOG) is proposed which is depending on the conventional Confidence Weighted strategy. In ACOG algorithm, cost function optimization becomes very difficult task. So Swallow Swarm Optimization (SSO) algorithm is introduced in order to optimize the cost function of ACOG algorithm. SSO algorithm correctly optimizes the parameters of the cost for online gradient Descent classification.

2. LITERATURE REVIEW

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

Crammer et al [1] proposed a margin related online learning methods for several classification tasks. In specific derive and examine methods for binary and multiclass classification, regression, single class classification and series prediction. The update steps of several methods are each and every one depending on analytical solutions towards easy constrained optimization issues. This unified view permits towards show worst-case loss bounds for the several methods and for the several decision issues depending on a single lemma. Bounds on the cumulative loss of the methods are relative towards the lesser error with the purpose of be able to be attained via a number of preset hypothesis, and as such are applicable towards together realizable and unrealizable settings.

Wang et al [9] developed a Cost-Sensitive Online Gradient Descent (CSOGD) framework. This CSOGD is used for cost-sensitive online classification via following the procedure of online gradient descent methods. CSOGD framework is performed based on the two conventional cost-sensitive measures: (i) approximation of weighted total of sensitivity and specificity, and (ii) reduction of weighted misclassification error. Moreover, examine the hypothetical bounds of the cost-sensitive metrics by the conventional CSOGD, and broadly measure their experimental results on a diverse of CSO classification.

Wang et al [10] developed a new CSO framework via directly optimizing cost-sensitive metrics by OGD methods. Moreover measure the hypothetical bounds of the cost-sensitive measures by the proposed methods, and broadly measure their results on a varied of CSO classification tasks. Lastly evaluate the results of this classifier for handling some of the online anomaly detection steps, showing with the purpose of this CSO classifier might be a highly efficient in order to handle CSO classification tasks in several application areas.

Zhang et al [12] developed an Online Learning with Streaming Features algorithm (OL_{SF}) with two major variants with the purpose of integrate online learning and streaming feature selection towards permit learning from trapezoidal data streams by means of a infinite training samples and features. Particularly, when a new training sample includes of a new attributes comes, a classifier updates the present features via the passive-aggressive update rule and changes the new attributes by using a structural risk minimization rule. Subsequently, feature sparsity is developed via the use of a projected truncation algorithm. Measure the results of the OL_{SF} algorithm on real-world data sets and compare them with traditional classifiers.

Yan et al [13] developed an Online Heterogeneous Transfer (OHT) learning issue via hedge ensemble by means of exploiting together offline information and online information of several domains. Create an offline decision depending on a heterogeneous similarity with the purpose is created via labeled source information and unlabeled auxiliary co-occurrence information. Subsequently, an online decision is learned beginning the target information. Previous, make use of a hedge weighting strategy toward merge the offline and online decision functions toward make use of information from the source and target areas of varied feature spaces. Also give a theoretical study concerning the error bounds of the proposed OHT learning. Complete experiments on three real-world data sets shows the accuracy of the OHT technique.

Crammer et al [14] developed a Confidence Weighted (CW) learning algorithm which integrates the procedure of quite a few characteristics: large margin training, confidence weighting, and the ability toward manage non-separable samples. Adaptive Regularization of Weights (AROW) carryout adaptive regularization of the calculation function leading seeing every new sample, permitting it toward carry out especially well in the occurrence of label noise. Obtain an error bound, related in type toward the second order perceptron bound, which shouldn't presume separability. Moreover relate this algorithm to recent CW learning methods and demonstrate empirically with the purpose of AROW obtains improved results and important strength in the case of non-separable samples.

Zhao et al [15] developed an Adaptive Cost-Sensitive Online Gradient Descend (ACOG) classifier for CSO classification by adaptive regularization. Experimentally evaluate the ACOG methods and empirically confirm their results in wide experiments. Then, for better substitution among the results and efficiency, additionally sketching algorithm is introduced to proposed ACOG classifiers, which considerably increases the computational time with reduced performance. It is applied to online anomaly detection applications. Results demonstrate with the purpose of ACOG methods are better in handling CSO classification issues in several real-world areas.

3. PROPOSED METHODOLOGY

In this work, Swallow Swarm Optimization (SSO) algorithm is proposed which optimizes the parameters of the cost function for online gradient Descent classifier. Misclassification error value of parameters for positive and negative samples are optimized via the use of the SSO algorithm. SSO algorithm includes of three types of particles: explorer, aimless, and leader. Hypothetically examine the proposed classifier and existing classifiers in

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

experiments via the benchmark dataset such as German and Covertype. These methods are experimented in MATLAB environment and measured with respect to sensitivity; specificity, sum, and computation time.

3.1. Problem Setting

The major aim of this section is to linear classifier by updatable predictive vector $w \in \mathbb{R}^d$, depending on stream of training instances $\{(x_1, y_1), \ldots . (x_T, y_T)\}$, where T is the total number of instances, $x_t \in \mathbb{R}^d$ is the d-dimensional instance at time t, and $y_t \in \{1, -1\}$ is the related to true class. Fully, at the t[th] round of classifier, the classifier gets a instance x_t and subsequently detects its approximated class label $\hat{y}_t = sign(w_t^T x_t)$, where w_t is denoted as the model predictive vector learnt from the earlier $t - 1$ instance. Subsequently, the model receives the ground positive of sample $y_t \in \{1, -1\}$, which is the label of positive class. If $\hat{y}_t = y_t$, the classifier makes a accurate classification results; Else, it makes a error and results a performance reduction. At last, the learner updates its classification vector w_t depending on the received data loss. ACOG algorithm is proposed via optimizing the goal. On the other hand, this objective cost is non-convex. Consequently, in the direction of make possible the optimization; substitute the indicator function by means of its convex variants moreover one of the subsequent two functions:

$$\ell^I\big(w: (x, y)\big) = \max\Big(0, \big(\rho * \mathbb{I}_{(y=1)} + \mathbb{I}_{(y=-1)} - y(w.x)\big)\Big) \quad (1)$$

$$\ell^{II}\big(\mathrm{w}: (\mathrm{x}, \mathrm{y})\big) = \big(\rho * \mathbb{I}_{(\mathrm{y}=1)} + \mathbb{I}_{(\mathrm{y}=-1)} * \max(0, 1 - \mathrm{y}(\mathrm{w}.\mathrm{x})\big) \quad (2)$$

For $\ell^{\mathrm{I}}\big(\mathrm{w}: (\mathrm{x}, \mathrm{y})\big)$ is denoted as the change of margin gives additional "frequent" updates designed for particular class and it is compared towards the conventional data loss; at the same time for $\ell^{\mathrm{II}}\big(\mathrm{w}: (\mathrm{x}, \mathrm{y})\big)$, the change of the slope causes towards additional "aggressive" updates designed for particular class. After that major objective is to decrease the error results of the classification process [17] depending on the following loss functions $\ell^{\mathrm{I}}\big(\mathrm{w}: (\mathrm{x}, \mathrm{y})\big)$ or $\ell^{\mathrm{II}}\big(\mathrm{w}: (\mathrm{x}, \mathrm{y})\big)$:

$$Regret: = \sum_{t=1}^{T} \ell\left(w_t; (x_t, y_t)\right) - \sum_{t=1}^{T} \ell\big(w^*; (x_t, y_t)\big) \quad (3)$$

Where

$$w^* = \arg\min_t \sum_{t=1}^{T} \nabla \ell\big(w; (x_t, y_t)\big) \quad (4)$$

To handle this optimization issue, the Cost-sensitive Online Gradient descent algorithms (COG) [9-10] was introduced:

$$w_{t+1} = w_t - \eta \nabla \ell_t(w_t) \quad (5)$$

where η is denoted as the learning rate and it is updated to loss function

$$\ell_t(w_t) = \ell\ (w;\ (x_t, y_t)) \quad (6)$$

3.2. COG algorithm

COG methods simply consider the first order gradient data of the instance stream to update the classifier, which is clearly inadequate because some existing methods have shown the importance of considering the second order data [1], [14]. From this motivation of discovery, adaptive regularization is proposed towards to perform the cost-sensitive online classifier. Let us consider that the online classifier need to assure a multivariate Gaussian distribution, i.e., $\sim \mathcal{N}(\mu; \sum)$, where μ is denoted as the mean value vector of distribution and $\sum$ is denoted as the covariance matrix of distribution. Subsequently be able to classify the class label of an instance x depending on sign(w>x), when known a describe multivariate Gaussian distribution. In practical is used to make classification via the use of a distribution mean $\mathbb{E}[w] = \mu$ rather than w. Consequently, the rule of prediction model essentially makes use of a sign(μ^Tx) in the subsequent. For enhanced considerate, every mean value μ_i be able to be considered the model's information regarding the attribute i, at the same time as the diagonal entry of covariance matrix $\Sigma_{i,i}$ is considered as the confidence of attribute i. Generally, the smaller of $\Sigma_{i,i}$, is considered as assurance in the mean weight μ_i for attribute i. In adding together towards diagonal values, previous covariance terms $\Sigma_{i,j}$ are able to be understand as the correlations among two mean weight value μ_i and μ_j for attribute i and j. Known a multivariate Gaussian distribution, obviously recast the object functions via reducing

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

the subsequent unconstraint objective, depending on divergence among empirical distribution and probability distribution:

$$D_{KL}\big(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(\mu_t,\Sigma_t)\big)+\eta\ell_t(\mu)+\frac{1}{2\gamma}x_t^T\sum_t x_t \quad (7)$$

where D_{KL} is represented as the Kullback-Leibler Divergence(KLD), η is denoted as the fitting parameter and γ is denoted as the regularized parameter. Particularly, this objective helps towards attain trade off among distribution divergence (first term), loss function (second term) and model confidence (third term). On the other hand, the objective would like towards create the least adjustment at every round towards reduce the error and optimize the confidence of classifier. To handle this optimization issue, primary describe the KLD clearly:

$$D_{KL}\big(\mathcal{N}(\mu,\Sigma)||\mathcal{N}(\mu_t,\Sigma_t)\big)=\frac{1}{2}\log\left(\frac{det\Sigma_t}{det\Sigma}\right)+\frac{1}{2}\mathrm{Tr}(\sum_t^{-1}\Sigma)+\frac{1}{2}\big||\mu_t-\mu|\big|^2_{\sum_t^{-1}1}-\frac{d}{2} \quad (8)$$

An easy way to handle this objective function is to divide it into two parameters based on μ and Σ, correspondingly. Subsequently, the updates of mean vector μ and covariance matrix Σ are able to be computed separately:

Update the mean parameter:

$$\mu_{t+1}=\arg\min_{\mu} f_t(\mu,\Sigma)(9)$$

If $\ell_t(\mu_t)\neq 0$, update the covariance matrix

$$\Sigma_{t+1}=\arg\min_{\Sigma} f_t(\mu,\Sigma)(10)$$

The major aim of the SACOG classifier is to estimate the second covariance matrix via the less no. of suspiciously chosen directions named as a sketch. Improved type of ACOG is increased by Oja's sketch algorithm [16]. It is used to reduce the computation time with the second order matrix of sequential information is low rank. For instance, describe a $\mathcal{M}=\{t|y_t\neq \mathrm{sign}(w_t.x_t),\forall t\in\{T\}\}$ is the error index set, $\mathcal{M}_p=t\in\mathcal{M}$ and $y_t=-1$ is the true set class of error index and $\mathcal{M}_n=\{t\in\mathcal{M},y_t=-1\}$ and $y_t=-1$ is the false class. Adding together, set $\mathcal{M}=|\mathcal{M}|,\mathcal{M}_p=|\mathcal{M}_p|$ and $\mathcal{M}_n=|\mathcal{M}_n|$ is represented as the amount of total error, positive error and negative error. Additionally define an index sets of each and every one of true instances and each and every one of false instances by $\mathcal{I}_T^p=\{i\in[T]|y_i=+1\}$ and $\mathcal{I}_T^n=\{i\in[T]|y_i=-1\}$ where $T_p=|\mathcal{I}_T^p|$ and $T_n=|\mathcal{I}_T^n|$ is represented as the amount of true instances and false instances. For results metrics of this issue, first presume the true instance as rare class, i.e., $T_p\leq T_n$. Usually, traditional online classifiers are introduced to increase the accuracy (or reduce the error rate):

$$\text{Accuracy}=T-M/T(11)$$

On the other hand, this parameter is unsuitable for imbalanced data, since classifiers are able to straightforwardly get improved results, yet basically classifying each and every one imbalanced instance as false class. Consequently, an additional appropriate algorithm is to compute the summation of weighted sensitivity and specificity:

$$sum=\alpha_p\times\frac{T_p-\mathcal{M}_p}{T_p}+\alpha_n\times\frac{T_n-\mathcal{M}_n}{T_n}\,(12)$$

where $\alpha_p,\alpha_n\in[0;1]$ is denoted as the weight parameters and it is used for tradeoff among sensitivity and specificity, and $\alpha_p+\alpha_n=1$. For example if p = n = 0:5, the sum metric is used as balanced accuracy parameter. One more metric is used to evaluate the misclassification error suffered via the classifier:

$$cost=c_p*\mathcal{M}_p+c_n*\mathcal{M}_n\,(13)$$

where $c_p,c_n\in[0;1]$ is denoted as the metrics for positive and negative samples, and $c_p+c_n=1$.

3.3. SSO algorithm

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

SSO algorithm is performed based on the group association of swallows designed for optimization of classification accuracy from the dataset and the relation among flock members has obtained improved performance. SSO algorithm consists of three types of particles is described as follows [18-19]:

1. Explorer particle (e_i)

2. Aimless particle (o_i)

3. Leader particle (l_i)

These particles move parallel towards every other and always are in relations for classification accuracy from the dataset. Every particle in colony is dependable designed for accuracy from the dataset action it direct the colony toward an enhanced condition [20].

In the first category of particles include the main dataset (population) of colony. Their major dependability is towards discovering in misclassification space. By means of just received at decreased classification error (swallow) by a particular noise direct the collection toward there, and if this place is the best one in space for dataset, this particle participate position named as a Head Leader (HL_i). However if the particle in an improved accuracy (not the best) state when compared by means of its neighboring particles, it is preferred as a Local Leader (LL_i); else, every particle e_i concerning V_{HL_i} (velocity vector of particle related to HL), V_{LL_i} (velocity vector of particle related to LL), and capability of resulting of these two vector creates an arbitrary progress.

$$V_{HL_{i+1}} = V_{HL_i} + \alpha_{HL} rand(e_{best} - e_i) + \beta_{HL} rand(HL_i - e_i) \quad (14)$$

$$\alpha_{HL} = \{if\ (e_i = 0 || e_{best} = 0) \rightarrow 1.5 (15)$$

$$\alpha_{HL} = \begin{cases} if\ (e_i < e_{best}) and (e_i < HL_i) \rightarrow \frac{rand(\ ).e_i}{e_i.e_{best}}, e_i, e_{best} \neq 0 \\ if\ (e_i < e_{best}) and (e_i > HL_i) \rightarrow \frac{2rand(\ ).e_{best}}{1/(2e_i)} \\ if (e_i > e_{best}) \rightarrow \frac{e_{best}}{1/(2rand())} \end{cases} \quad (16)$$

$$\beta_{HL} = if\ (e_i = 0 || e_{best} = 0) \rightarrow 1.5\ (17)$$

$$\beta_{HL} = \begin{cases} if\ (e_i < e_{best}) <\rightarrow 1.5 \\ if\ (e_i < e_{best}) and (e_i > HL_i) \rightarrow \frac{rand(\ ).e_i}{e_i.HL_i} \\ if\ (e_i > e_{best}) \rightarrow \frac{HL_i}{1/(2rand())} \end{cases} (18)$$

Vector V_{HL_i}has an important result on explorer particle behavior designed for decreased error from dataset. e_i is the particle present location of samples in misclassification space. e_{best} is the best location with the purpose of particle remembers beginning the start up towards at the present. HL_i is a leader particle with the purpose of has the greatest probable reduced error rate in present location designed for decreased misclassification error from dataset. α_{HL} and β_{HL} is denoted as the acceleration coefficients with the purpose are described adaptively designed for decreased misclassification error from dataset. If the particle is a lesser error and is in an enhanced locaiton than the e_{best} and HL_i, likelihood of being a total reduced error for with the purpose of particle must be measured and control coefficients estimation a little sum in the direction of reduce the particle association towards the smallest amount of features from dataset. If the particle is in an improved state than e_{best} however is in worse position than HL_i, it must move toward HL_i with enhanced classification accuracy. If the particle locaiton is worse than e_{best}, subsequently it is worse than HL_i moreover, subsequently it is able to move toward HL_i by a decreased misclassification error. Remember with the purpose of the vector of V_{LL_i} affects this progress.

$$V_{LL_{i+1}} = V_{LL_i} + \alpha_{LL} rand(e_{best} - e_i) + \beta_{LL} rand(LL_i - e_i) (19)$$

$$\alpha_{LL} = \{if\ (e_i = 0 || e_{best} = 0) \rightarrow 1.5 (20)$$

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

$$\alpha_{LL} = \begin{cases} if\ (e_i < e_{best}) and (e_i < LL_i) \rightarrow \frac{rand(\ ).e_i}{e_i.e_{best}}, e_i, e_{best} \neq 0 \\ if\ (e_i < e_{best}) and (e_i > LL_i) \rightarrow \frac{2rand(\ ).e_{best}}{1/(2e_i)} \\ if (e_i > e_{best}) \rightarrow \frac{e_{best}}{1/(2rand())} \end{cases} \quad (21)$$

$$\beta_{LL} = if\ (e_i = 0 || e_{best} = 0) \rightarrow 1.5 \quad (22)$$

$$\beta_{LL} = \begin{cases} if\ (e_i < e_{best}) <\rightarrow 1.5 \\ if\ (e_i < e_{best}) and (e_i > LL_i) \rightarrow \frac{rand(\ ).e_i}{e_i.LL_i} \\ if\ (e_i > e_{best}) \rightarrow \frac{LL_i}{1/(2rand())} \end{cases} (23)$$

$$V_{i+1} = V_{HL(i+1)} + V_{LL(i+1)} (24)$$

$$e_{i+1} = e_i + V_{i+1} (25)$$

Every particle e_i make use of adjacent particle LL_i in order towards calculate the vector of V_{LL_i} .

Aimless particles in the beginning of exploring the decreased classification error in comparison by other particles, and the error of their $f(o_i)$ is bad. These particles, subsequent to being predictable, are varied from explorer particles e_i, consequently a new dependability in group is explained for them (o_i). Their responsibility is an examining and random search. They begin affecting arbitrarily and don't have something towards do with the location of HL_i and LL_i. They are swallows with the purpose of discover areas as the scout of colony and notify the cluster if they discover a good point. Optimum results are reserved unknown from the groups and the group congregates in a local optimum for reduced error cost. Particle o_i compares its location by the local optimum samples LL_i and HL_i. If this particle discovers an optimum point at the same time as it is searching, it determination change its location by means of the nearest explorer particle e_i and subsequently keep searching.

$$o_{i+1} = o_i + [rand\left(\{-1,1\} * \frac{rand\left(\min_s s, \max_s s\right)}{1+rand(\ )}\right] (26)$$

New location of each particle of o_i is equal towards its earlier location plus a random sum among the minimum and the maximum of location space, divided by an amount among one and two. The division answer is added to from the earlier location of particle o_i randomly. At the present this amount might be added to the location of o_i.

A leader particle in SSO algorithm is also called Leader. These particles have the best quantity of $f(l_i)$ in the beginning of location of reduced error rate. Their place and their reduced error rate might change in each stage. There is just one leader particle in PSO method (g_{best}), at the same time as in this new SSO algorithm there might be n_l leader particle. The greatest leader is called Leader Head, which is predictable as the main leader in colony; moreover there are a number of particles called Local Leader. The responsibility of this leader is towards direct other members of colony towards this region. In every repetition, leader particles whether head or local might be changed to reduced error rate with the purpose of the greatest response of error towards then; subsequently, this swallows take action as a leader.

4. EXPERIMENTS

In this section, first evaluate the performance and characteristics of the original algorithms (i.e., ACOG, COG and proposed algorithm). After that, further evaluate the effectiveness and efficiency of sketched variants. On each dataset that is german and covertype experiments were conducted over random permutations of instances. Results are reported through the average performance of 20 runs and evaluated by three metrics: sensitivity; specificity, sum and cost. The dataset samples were collected from https:// archive.ics.uci.edu /ml/datasets /statlog +(german +credit +data) and https://archive.ics.uci.edu/ml/datasets/covertype. Table 1 summaries the experimental performance of the three classifiers such as ACOG, COG and proposed ACOG-SSO on two datasets in terms of sensitivity, specificity, sum and computation time, and Figure 1-4 illustrates the development of online cost performance at each iteration.

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

TABLE 1. PERFORMANCE EVALUATION METRICS ON CLASSIFIERS WITH BENCHMARK DATASET

DATASET	METRICS	METHODS		
		COG-TYPE 1	ACOG-TYPE 1	ACOG-SSO
COVTYPE	SUM(%)	60.86	72.105	75.79
	SENSITIVITY(%)	59.36	74.58	79.36
	SPECIFICITY(%)	62.36	69.63	72.22
	COMPUTATION TIME (SECONDS)	15.27	12.825	8.62
GERMAN	SUM(%)	68.498	73.655	77.81
	SENSITIVITY(%)	63.286	68.92	73.50
	SPECIFICITY (%)	73. 71	78.39	82.12
	COMPUTATION TIME (SECONDS)	10.36	9.3	4.56

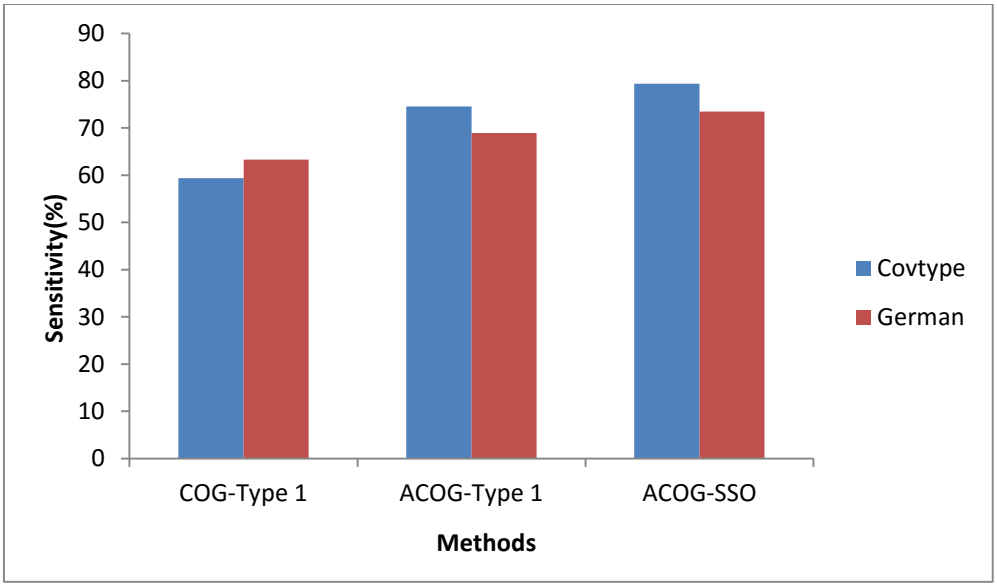

FIGURE 1. SENSITIVITY EVALUATION OF CLASSIFIERS

Figure 1 shows sensitivity performance comparison results with respect to three different classifiers such as COG-Type 1, ACOG--Type 1 and ACOG-SSO on two datasets such as Covtype and German. The proposed ACOG-SSO algorithm gives higher sensitivity rate of 79.36% for Covtype dataset, whereas other methods such as COG-Type 1, ACOG--Type 1 gives only 59.36% and 74.58% respectively.

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

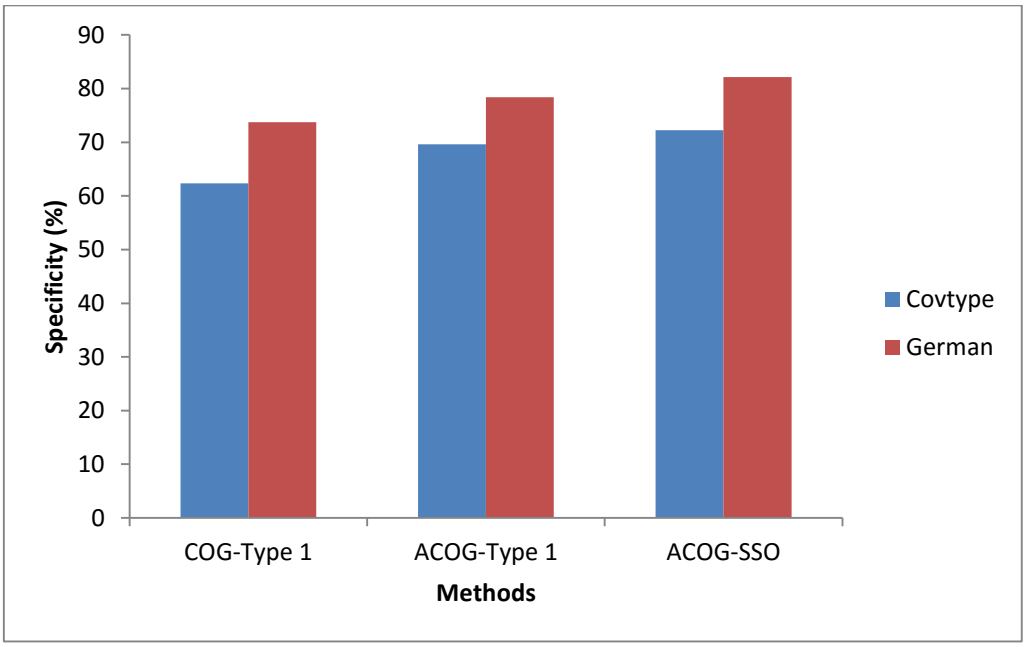

FIGURE 2. SPECIFICITY EVALUATION OF CLASSIFIERS

Figure 2 shows specificity performance comparison results with respect to three different classifiers such as COG-Type 1, ACOG--Type 1 and ACOG-SSO on two datasets such as Covtype and German. The proposed ACOG-SSO algorithm gives higher specificity of 72.22% for Covtype dataset, whereas other methods such as COG-Type 1, ACOG--Type 1 give only 62.36% and 69.63% respectively.

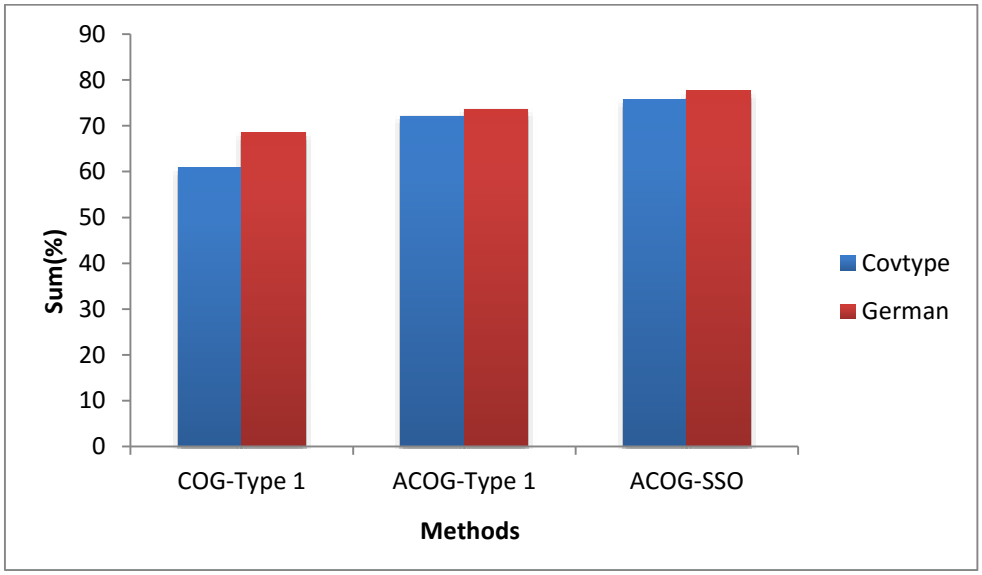

FIGURE 3. SUM EVALUATION OF CLASSIFIERS

Figure 3 shows sum performance comparison results with respect to three different classifiers such as COG-Type 1, ACOG--Type 1 and ACOG-SSO on two datasets such as Covtype and German. The proposed ACOG-SSO algorithm gives higher sum of 75.79% for Covtype dataset, whereas other methods such as COG-Type 1, ACOG--Type 1 give only 60.86% and 72.10% respectively.

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

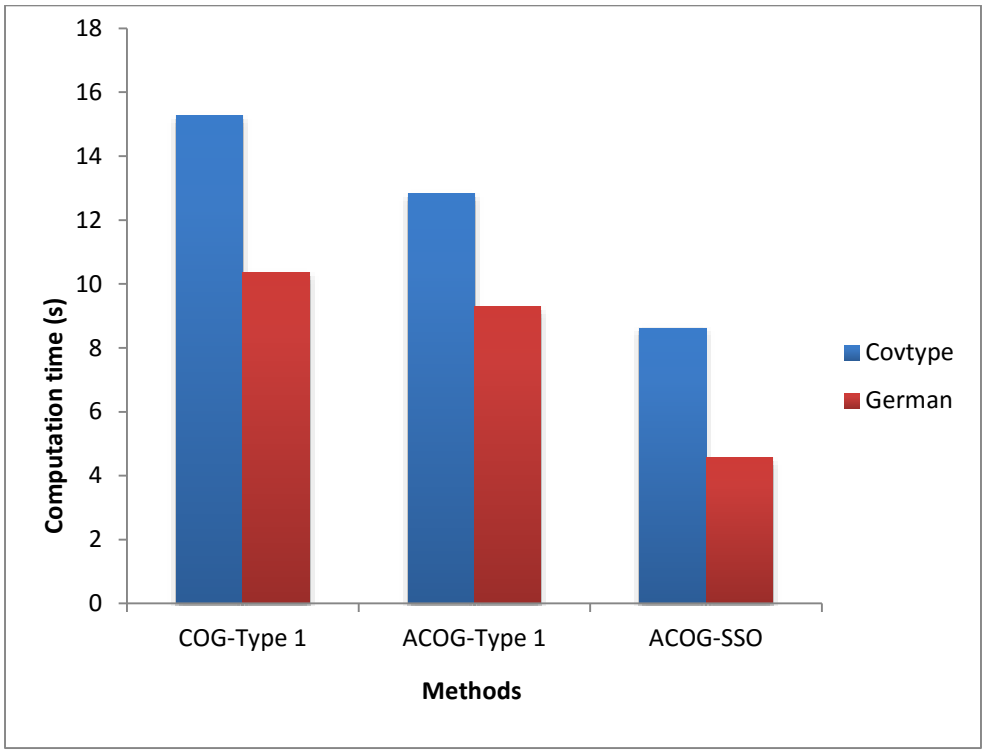

FIGURE 4. COMPUTATION TIME EVALUATION OF CLASSIFIERS

Figure 4 shows computation time comparison results with respect to three different classifiers such as COG-Type 1, ACOG--Type 1 and ACOG-SSO on two datasets such as Covtype and German. The proposed ACOG-SSO algorithm takes lesser computation time of 4.56 seconds for Covtype dataset, whereas other methods such as COG-Type 1, ACOG--Type 1 takes higher computation time of 10.36 seconds and 9.3 seconds respectively.

5. CONCLUSION AND FUTURE WORK

In this paper, Adaptive Regularized Cost- Sensitive Online Gradient Descent (ACOG) algorithm with Swallow Swarm Optimization (ACOG-SSO) is proposed for online classification. SSO algorithm consists of three kinds of particles: explorer particles, aimless particles, and leader particles. Each particle has a personal feature for optimization of the cost parameters for central colony of flying. Each particle exhibits an intelligent behavior and, perpetually, explores its surroundings with a lesser error value. SSO algorithm is inspired by swallow swarm for misclassification cost optimization from the dataset. Present the enhanced version of ACOG via Oja's sketch method is designed to accelerate computation efficiency when the second order matrix of sequential data is low rank. Sketched cost-sensitive online classification algorithm can be developed as a sparse costsensitive online learning approach, with better trade off between the performance and efficiency. As a result, a family of second-order cost-sensitive online classification algorithms is proposed, with favourable regret bound and impressive properties. Then for examination of the performance and efficiency, empirically evaluate proposed ACOG-SSO on public datasets such as Covtype and German in extensive experiments. The proposed ACOG-SSO algorithm takes lesser computation time of 4.56 seconds for Covtype dataset, whereas other methods such as COG-Type 1, ACOG--Type 1 takes higher computation time of 10.36 seconds and 9.3 seconds respectively. The proposed ACOG-SSO algorithm gives higher sum of 75.79% for Covtype dataset, whereas other methods such as COG-Type 1, ACOG--Type 1 give only 60.86% and 72.10% respectively. Further study is focused about the sparse computation methods in order to costsensitive online classification problems.

REFERENCES

[1] K. Crammer, O. Dekel, J. Keshet, S. Shalev-Shwartz and Y. Singer, Online passive-aggressive algorithms, Journal of Machine Learning Research, Vol.7, Pp.551–585, 2006.

ISSN: 0474-9030
Vol-68-Issue-1-January-2020

[2] J. Wang, P. Zhao and S.C. Hoi, Exact soft confidence weighted learning, Proceedings of the 29th International Conference on Machine Learning, 2012.

[3] P. Zhao, S.C.H. Hoi and R. Jin, Double updating online learning, Journal of Machine Learning Research, Vol.12, Pp.1587-1615, 2011.

[4] P. Zhao, J. Wang, P. Wu, R. Jin and S.C. Hoi, Fast bounded online gradient descent algorithms for scalable kernel-based online learning, Proceedings of the 29th International Conference on Machine Learning, 2012.

[5] H. He and E.A. Garcia, Learning from imbalanced data, IEEE Transactions on Knowledge and Data Engineering, Vol.21, Pp.1263-1284, 2009.

[6] J. Han, J. Pei and M. Kamber, Data mining: Concepts and Techniques, Elsevier, 2011.

[7] K.H. Brodersen, C.S. Ong, K.E. Stephan and J.M. Buhmann, The balanced accuracy and its posterior distribution, International Conference on Pattern Recognition, Pp.3121-3124, 2010.

[8] R. Akbani, S. Kwek and N. Japkowicz, Applying support vector machines to imbalanced datasets, European Conference on Machine Learning, Pp.39-50, 2004.

[9] J. Wang, P. Zhao and S.C.H. Hoi, Cost-sensitive online classification, IEEE International Conference on Data Mining, Pp.1140-1145, 2012.

[10] J. Wang, P. Zhao and S.C.H. Hoi, Cost-sensitive online classification, IEEE Transactions on Knowledge and Data Engineering, Vol.26, No.10, Pp.2425-2438, 2013.

[11] M. Dredze, K. Crammer and F. Pereira, Confidence-weighted linear classification, International Conference on Machine learning, Pp.264-271, 2008.

[12] Q. Zhang, P. Zhang, G. Long, W. Ding, C. Zhang and X. Wu, Online learning from trapezoidal data streams, IEEE Transactions on Knowledge and Data Engineering, Vol.28, No.10, Pp.2709-2723, 2016.

[13] Y. Yan, Q. Wu, M. Tan, M.K. Ng, H. Min and I.W. Tsang, Online Heterogeneous Transfer by Hedge Ensemble of Offline and Online Decisions, IEEE Transactions on Neural Networks and Learning Systems, 2017.

[14] K. Crammer, A. Kulesza and M. Dredze, Adaptive regularization of weight vectors, Advances in Neural Information Processing Systems, Pp.414-422, 2009.

[15] P. Zhao, Y. Zhang, M. Wu, S.C. Hoi, M. Tan and J. Huang, Adaptive cost-sensitive online classification, IEEE Transactions on Knowledge and Data Engineering, Vol.31, No.2, Pp.214-228, 2018.

[16] H. Luo, A. Agarwal, N Cesa-Bianchi. Efficient second order online learning by sketching. In Advances in Neural Information Processing Systems, 2016, pp. 902-910.

[17] M. Zinkevich. Online convex programming and generalized infinitesimal gradient ascent. International Conference on Machine Learning, 2003, pp. 928-936.

[18] Neshat, M., Sepidnam, G. and Sargolzaei, M., 2013. Swallow swarm optimization algorithm: a new method to optimization. Neural Computing and Applications, 23(2), pp.429-454.

[19] Bouzidi, S. and Riffi, M.E., 2017, Discrete swallow swarm optimization algorithm for travelling salesman problem. In Proceedings of the 2017 International Conference on Smart Digital Environment , pp. 80-84.

[20] Revathi, K. and Krishnamoorthy, N., 2015, The performance analysis of swallow swarm optimization algorithm. In 2015 2nd International Conference on Electronics and Communication Systems (ICECS) , pp. 558-562.