Hybrid feature selection by combining f-score & BPSO

Gopal kumar
Department of Information technology National institute of Technology ,Patna

A R T I C L E   I N O F
Keywords:
Feature selection
F-score
BPSO
Support vector machine
HAR (human activity
Recognition)

a b s t r a c t
Feature selection is a technique to find the most relevant feature that helps to increase the speed of computation and the accuracy. Human activity recogniser sensor generate the lots of data with lots of feature when this data is used for classification the classifier may be or definitely give high error rate.Hence in this paper we use the hybrid framework to obtain the optimal no of feature on the HAR* data set.
It is observed from our experiment F-score and BPSO hybrid framework has given best optimal no of feature and good accuracy .

R E L A T E D   W O R K

Feature selection method has been used in the classification problem in order to select the optimal feature set that makes the classifier more accurate and faster. Some specific problem are always proceeds with large no of features. For instance HAR data set is one of them which contains the large no of feature.here we are focus on the hybrid feature selection based on F-score and BPSO.

<u>F-score</u> :-

F-score is used to calculate the discriminative ability of each feature with the higher F-score have better separation ability in the the classification problem .

F-score is define by following equation.

$$F(i) \equiv \frac{\left(\bar{x}_i^{(+)} - \bar{x}_i\right)^2 + \left(\bar{x}_i^{(-)} - \bar{x}_i\right)^2}{\frac{1}{n_+ -1}\sum_{k=1}^{n+}\left(x_{k,i}^{(+)} - \bar{x}_i^{(+)}\right)^2 + \frac{1}{n_1 -1}\sum_{k=1}^{n-}\left(x_{k,i}^{(-)} - \bar{x}_i^{(-)}\right)^2}$$

Where $\bar{x}(+)$,$\bar{x}(-)$ and $\bar{x}$ are the average of the ith feature of positive sample Negative-sample and whole data set respectively , and here k is define as the kth sample of positive and negative sample corresponding to $\bar{x}(+)$ and $\bar{x}$ (-).

F-score can examine the discriminative ability of each individual feature .if The F-score is high that show the feature has stronger discriminative ability.

BPSO :-

L Cervante et al.exploited BPSO for feature selection purpose. BPSO follows swarm behaviour to find optimal solution. Particles in the swarm are considered as possible subsets of the genes for the given problem. BPSO controls the velocity of the particle depending upon the value of local best solution and global best solution. Local best solution is the maximum fitness value achieved by respective particle, whereas global best is the maximum fitness among all the particles in the swarm
particle is updated based upon velocity value as shown in below equation .This approach ensures survival of the fittest genes in the particle,i.e. the position of the particle will be updated if it is not contributing to find better solution.BPSO differs from PSO in type of values generated for the next position of the particle, it generates only binary values. It uses sigmoid function to restrict values to 0 or 1. Hence genes which are not contributing will be given value as 0, that means that particular gene will not be considered while evaluating the solution of the problem. BPSO finds optimal solution by updating current position depending upon velocity value as per the following equations:

$$x_{id}^{t+1} = x_{id}^{t} + v_{id}^{t+1}$$

$$v_{id}^{t+1} = w * v_{id}^{t} + c_1 * r_{1i} * (p_{id} - x_{id}^{t}) + c_2 * r_{2i} * (p_{gd} - x_{id}^{t})$$

Where, current position of the particle i is represented by x tid , dD denotes the dimension in the search space, dimensionality of features is

represented as D, t denotes the iteration number in the search process. c 1 and c 2 are acceleration constants. w is inertia weight. r 1i and r 2i are random values uniformly distributed in [0, 1]. p id and p gd represents the elements of local best and global best in the dimension. BPSO restricts values of x id ,
p id and p gd to 0 or 1 using sigmoid function as given below:

$$x = \begin{pmatrix} 1, & \text{if rand}() < s(v_{id}) \\ 0, & otherwise \end{pmatrix}$$

-(1.0)

Where ,

$$s(v_{id}) = \frac{1}{1 + e^{-v_{id}}}$$

- (1.1)

Sigmoid function s(v id) is used to distribute velocity v id value in range (0, 1). rand() is random number generated from a uniform distribution in [0,1].

Feature selection using BPSO

BPSO can be used for feature selection problem, as feature selection problem is discrete in nature. Position of the particle is represented in binary where, 1 represents feature considered and 0 represents not considered. For feature selection problem, p gd of particle i is evaluated using predictor performance of the particle as the evaluation measure. In proposed framework we have used hold out method of validation with 71% training and 29% testing to compute predictor performance.
In each iteration t, particle i is evaluated using evaluation function and local best p id and global best p gd values are updated using velocity and distance Equation (1.0) and(1.1) restrict position value to binary using

sigmoid function. After completion of t iterations, particle associated with value, gives near optimal feature subset as output.

F R A M E W O R K

For performing optimal feature selection, a frameworks is proposed . .Framework contains two stage . The first stage of framework use F-score and second stage contains BPSO.

Framework :- F-score BPSO

Framework is a wrapper followed by wrapper hybrid model where F-score is used as wrapper-1 and BPSO is used as wrapper -2

When set of feature is given to F-score framework F-score rank the all feature of positive sample according to equation -1 and and above certain threshold t are selected and supplied as input to the iterative BPSO based on wrapper for further optimization.

The framework's execution is describe in the following steps:

1) Step 1:
   A dataset with K feature is supplied as input to stage 1 of the framework.
2) step2:
   F-score works as following step

   a) Input is K feature with 'c' no of class ranking feature (using equation□-1)taking 1st class as positive and all other as negative.
   b) Repite for all class
   c) Study the curve
   d) And define the threshold and select the feature > threshold value ("f1,f2,f3,f4,f5,f6 feature for respective class)

e) Taking the intersection($f1 \cap f2 \cap f3 \cap f4 \cap f5 \cap f6$)
f) 'F' no of feature get from step "e"

3. Step 3(stage 2- iterative BPSO):
stage 2 uses iterative BPSO as wrapper 2 which takes the F number of features
supplied from stage 1 as it's particles.
Particle is random set of features selected that can be represented as $p = \{x_1, x_2, x_j, \ldots, x_F\}$,where x_j represents feature j th among F features.
BPSO uses a collection of particles called as swarm of particles, which can be represented as $S = \{p_1, p_2, p_j, \ldots, p_F\}$.
Algorithm for stage 2 can be given as:
• condition 1:
fitness value of p_{gd} of i th iteration is less than (i − 1) th iteration
• condition 2:
number of features considered by p_{gd} of i th iteration is greater than (i − 1) th
iteration for the same fitness value of p_{gd}
repeat until(condition 1 or condition 2)

//repeat 1
repeat until(number of iterations is equal to t)// repeat 2
for all the particles in the swarm do

(a) Construct dataset with reduced dimension k_0 , where k_0 is the number of features considered by the particle

(b) Train the classifier with reduced data using hold out method to find out performance of the particle (i.e. fitness value of the particle)

(c) Update values of x_{id} , v_{id} , p_{id} and p_{gd} according to equation (4),(5),(6) and (7).
end for

end repeat // repeat 2

return (p gd and it's fitness value)

end repeat

return (optimal feature subset)

4. Step 4:

Return optimal feature subset

// repeat 1

RESULT

<u>43 feature are selected:</u>

1) "fBodyAccJerk-std()-Y\n",
2)"tBodyAcc-min()-X\n' ,
3)tBodyAcc-min()-Y\n',
'4)tBodyGyroJerk-correlation()-X,Y\n',
'5)tBodyGyroJerk-correlation()-X,Z\n',
'6)fBodyAcc-sma()\n',
7)'angle(tBodyAccMean,gravity)\n',
'8)tBodyGyroMag-mean()\n',
9)tGravityAcc-energy()-X\n',
10)fBodyAccJerk-meanFreq()-Z\n',
11)'fBodyAccJerk-meanFreq()-X\n',
12)fBodyAccJerk-meanFreq()-Y\n',
13)'fBodyAcc-meanFreq()-Y\n',
'15)14)fBodyAcc-meanFreq()-Z\n',
'16)tBodyAccJerkMag-arCoeff()2\n',
'17)tBodyAccJerkMag-arCoeff()1\n',
'18)fBodyBodyAccJerkMag-mean()\n',
'19)tBodyGyroJerk-entropy()-Y\n',
'20)angle(tBodyAccJerkMean),gravityMean)\n',
'21)fBodyAcc-mad()-Y\n',
'22)fBodyGyro-skewness()-Y\n',
'23)fBodyBodyGyroJerkMag-entropy()\n',
'24)tBodyGyroMag-arCoeff()1\n',
'25)tBodyGyroJerk-arCoeff()-Y,1\n',
'26)tGravityAcc-min()-Y\n',

'27)tBodyGyroJerk-arCoeff()-Y,3\n',
'28)fBodyGyro-meanFreq()-Z\n',
'29)tBodyGyroJerk-arCoeff()-Z,1\n',
30)'tBodyGyro-arCoeff()-Z,2\n',
31)'tBodyGyro-arCoeff()-Z,4\n',
'32)fBodyAcc-std()-Y\n',
'33)tBodyGyroJerk-arCoeff()-Z,4\n',
'34)tBodyGyro-entropy()-Z\n',
'35)tBodyGyro-entropy()-X\n',
'36)tBodyGyro-entropy()-Y\n',
'37)fBodyGyro-entropy()-X\n',
'38)tBodyGyroJerk-arCoeff()-X,2\n',
'39)fBodyAccMag-sma()\n',
'40)tBodyAccMag-arCoeff()4\n',
41)'angle(Y,gravityMean)\n',
42)tBodyAccMag-arCoeff()3\n',
'43)tBodyAcc-arCoeff()-Z,2\n']

Classification report: (43 feature)

class	precision	recall	f1-score	support
LAYING	1.00	1.00	1.00	537
SITTING	0.95	0.89	0.92	491
STANDING	0.91	0.95	0.93	532
WALKING	0.99	0.98	0.99	496
WALKING_DOWNSTAIRS	0.98	0.98	0.98	420
WALKING_UPSTAIRS	0.98	1.00	0.99	471
avg / total	0.97	0.97	0.97	2947

Reference -