Extending the Nelder-Mead Algorithm for Feature Selection from Brain Networks

Reshma Kar1, Amit Konar1, Aruna Chakraborty2, Anca L. Ralescu3 and Atulya K. Nagar4

1Jadavpur University, Kolkata, India
2St. Thomas College of Engineering & Technology, Kolkata, India
3University of Cincinnati, USA
4Liverpool Hope University, Liverpool, UK

rkar317@gmail.com, konaramit@yahoo.co.in, arunachakraborty.stcet@gmail.com, Anca.Ralescu@uc.edu, nagara@hope.ac.uk

Abstract—Centrifugation is often applied in laboratories and industries to increase the effective gravity on a particle and hence make it sediment faster. Based on this principle, one may extend the existing optimization techniques, which are driven by only gravitational force (objective values of discovered best solutions), and do not consider application of centrifugal force for faster convergence. We extended the Nelder-Mead’s simplex algorithm, by applying an exponentially decaying centrifugal force on each of the computed vertices of the simplex. The proposed centrifugation technique was also applied on other optimization algorithms including differential evolution and gravitational search algorithms. It was seen that application of centrifugal force indeed enhanced the objective values obtained by of all the tested evolutionary algorithms. The comparative performance of the extended Nelder-Mead Algorithm was found to be better among all the tested algorithms. The algorithms were compared on the basis of the best obtained objective value after a fixed number of objective function evaluations (here 20 times the problem dimension). Testing was performed in the real world problem of EEG feature selection (from brain networks), for the classification of memory encoding versus recall using SVM. The average classification accuracy was found to be high (89.97%).

Keywords—Centrifugation, Nelder-Mead Algorithm, Simplex, Optimization, Electroencephalogram, Brain Networks.

I. INTRODUCTION

Optimization problems are often multi-dimensional and bring an increasing complexity with increasing dimension. Unlike many other population based optimization algorithms [1],[2], the simplex algorithm proposed by Nelder and Mead [3] considers the dimension of the problem space while initializing its initial guesses/population, thus molding itself to the computational complexity of the problem. Interestingly, the Nelder-Mead (NM) algorithm is considered as an evolutionary algorithm for its specific characteristics of population based evolutionary optimization with special selection and reproduction operators [4]. The NM algorithm initializes a closed polygon in an n dimensional (problem) space which has n+1 vertices. Such a closed polygon is called a simplex as it is the polygon with the minimal number of possible edges in that dimension. The n+1 vertices comprise the population of initial guesses for solving the optimization problem. The NM algorithm then explores solutions by reflection, expansion, contraction and shrinking of the simplex based on the relative objective values of the different vertices.

In our experiments, the NM algorithm outperformed its popular counterparts on the given set of optimization problems. We also applied the principle of centrifugation to each of the tested algorithms and discovered that the principle can be employed to enhance the objective values obtained by the respective algorithms, including the traditional NM algorithm.

In laboratory and industrial settings, centrifugation is often performed in order to increase the rate of sedimentation. The working principle behind this is simple. At any point of time, a constant amount of gravitational force acts on a sample (for example, a viscous mixture in a test-tube). By moving the sample in circular manner at high speed, the effective gravitational force on each particle of the sample is increased. This ensures that the particles with more mass settle down faster. Although centrifugation is a common technique, to the best of our knowledge, the principle behind this has never been employed to accelerate the convergence of an optimization problem. In this paper, we have outlined this interesting application of centrifugal force to a selective set of optimization problems. Our experiments reveal that application of centrifugal force indeed enhances convergence in optimization problems.

In order to simulate the centrifugal force, we added an exponentially decaying number to solution(s) computed by traditional algorithms. This sum was subjected it to non-linear operation followed by truncation, which simulates a circular movement in the computed solution space.

The proposed technique was tested on the real world problem of Electroencephalogram (EEG) based brain network feature selection for the classification of memory encoding versus recall. The problem of decoding the state of learning/memory of a person has significant implications in rehabilitative applications as well as diagnostic applications for dementia related disorders. In this work, feature selection was performed on brain network features.

A brain network created by considering n EEG electrodes, typically has \(^nC_2\) elements indicating the pairwise connectivity (edges) between each of the n electrodes (brain network nodes). More number of electrodes potentially means more precision in decoding mental tasks. Unfortunately, more
electrodes mean a larger feature space which can inversely impact the recognition rate of a classifier. Thus one may employ feature selection schemes to select the best connectivity edges in the brain and reject the others.

Multiple criteria were kept in mind while developing the feature selection heuristic proposed in [6]. The fundamental idea is to minimize inter-class feature variance and maximize intra-class feature variance. These criteria were combined to define a single-objective heuristic which was attempted to be minimized by optimization as it is a complex and non-linear problem. Here minimization was performed by an extended variant of NM Algorithm which we shall hereafter refer to as Centrifugal Nelder-Mead (CNM) algorithm. The NM algorithm was chosen to be extended, as it outperformed its counterparts on the selected problem set.

Section II discusses the NM Algorithm in detail; Section III outlines the proposed extension of the NM algorithm; In Section IV a case study is described in which the feature selection from brain networks is performed for the purpose of classification of memory encoding versus recall.

II. THE THE NELDER-MEAD SIMPLEX ALGORITHM

A. The Nelder-Mead Simplex Algorithm

A simplex is a polygon with the least number of vertices in an n dimensional space. In order to form a simplex in an n dimensional space, one needs at least n+1 vertices. The Nelder-Mead algorithm randomly initializes a simplex of solutions in the search space, and updates the vertices in order to reach the optima. The simplex algorithm is one of the best known optimization algorithms[7].

Let us consider an n dimensional solution space. The NM algorithm starts by initializing n+1 vertices in the search space in order to create a simplex. Let the i-th vertex denoted by w_i be associated with a fitness f_i. The geometrical transformations of the simplex are illustrated for a 2 dimensional solution space in Fig. 1.A popular variant of the Nelder-Mead algorithm [7], is outlined below.

Steps of The Nelder-Mead Algorithm (NMA)

1. **Sort:** Arrange all the vertices w_i (guesses for selected combination of features in desired dimension) according to ascending order of objective function values such that

   \[ f_1 < f_2 < \ldots < f_{n+1} \]

   where \( f_i \) is the function value of the i-th vertex \( w_i \)

2. **Reflect:** Reflect the simplex away from the worst point as demonstrated in Fig. 1(a), by the following computations.
   a) Compute centroid \( c \) of the \( n \) best points

   \[
   c = \frac{\sum_{i=1}^{n} w_i}{n}
   \]

   b) Compute reflected point \( x_r \)

   \[
   x_r = c + R(c - x_{n+1})
   \]

   where \( R \) is the reflection coefficient, generally \( R=1 \).

3. **Expand:** If reflected point is better than the best point obtained so far, expand the simplex along the direction of the reflected point. This causes the algorithm to speed up towards potential minima (Fig. 1(b)).

   If \( f_r < f_1 \)
   a) Compute expanded point \( x_e \)

   \[
   x_e = c + E(x_r - c)
   \]

   where \( E \) is the expansion Coefficient, generally made equal to 2.

   Else

   \[
   x_{n+1} = x_r
   \]

4. **Contract:** If reflected point is worse than the second worst point obtained so far, it means that the simplex was reflected too far and must be projected nearer to the centroid \( c \) (Fig. 1(c)).

   If \( f_r > f_n \)
   a) If \( f_n < f_r < f_{n+1} \)

   Compute contracted point \( x_c \) outside the simplex.

   \[
   x_c = c + C(x_r - c)
   \]

   If \( f_c < f_r \)

   \[
   x_{n+1} = x_c
   \]

   Else

   Perform Shrink operation

   End If
b) Else if \( f_{n+1} < f_r \)
Compute contracted point \( x_c \) inside the simplex.
\[
x_c = c + C(x_{n+1} - c)
\]
If \( f_c < f_r \)
\[
x_{n+1} = x_c
\]
Else
Perform Shrink operation
End If

where \( C \) is the contraction coefficient, generally made equal to 1/2.

5. Shrink: Following reflection and contraction, if no feasible solutions are generated, shrink all the vertices of the simplex towards the best known solution (Fig. 1 (d)).

a) Compute \( n \) new vertices
For \( j=1, 2, ..., n \)
\[
x_j = c + S(x_j - x_1)
\]
End For

Here, generally \( S=1/2 \).

Repeat from Step 1 till a convergence criterion is not met.

III. THE PROPOSED CENTRIFUGAL NELDER-MEAD ALGORITHM

The Centrifugal Nelder-Mead (CNM) adds an exponentially decaying centrifugal force to the computed solutions of the Nelder-Mead Algorithm (NMA) and then truncates solutions within bounds. The decay of centrifugal force ensures that the actual gravitational force will gain dominance during the latter part of the algorithm, which will increase exploitation. During the initial phase, the centrifugal force is high thus resulting in increased exploration. The pseudo code for the proposed algorithm is given below. Two new procedures are added to the NM algorithm namely Centrifuge and Truncate. These modules are briefly described and then the pseudo code for the centrifugal NM (CNM) algorithm is then outlined.

A. Centrifuge

The centrifuge module applies a force in the direction of the upper bound, to the computed solution vectors. This may increase the solution values above the permitted range and hence the resultant vector is subjected to a non-linear operation which is fundamental in simulating a circular force. It is also desirable to slowly decrease the speed of centrifugation before stopping it. In order to simulate this, the force that was added along the direction of the upper bound is made to decay exponentially with increasing number of function evaluations (here \( t \)). Thus, the decaying force is computed as \( e^{-kt} \), where \( k \) determines the rate at which this decay takes place. Here \( k \) decreases as the dimension of the problem increases. This is because a large search space needs more exploration and thus the centrifugal force must act for a longer time.

B. Truncate

The Truncate module brings the weight vectors within bounds the \((lb, ub)\). Where \( lb \) is the lower bound and \( ub \) is the upper bound. It works by a Max-Min operation which selects the respective boundary value if the computed weight vector crosses it.

Output: Centrifuged population vector \( w_i \)

Begin:
If \( (w_i + e^{-kt}) > 1 \)
\[
w_i = (w_i + e^{-kt}) = (w_i + e^{-kt})
\]
Procedure: Centrifugal Nelder-Mead (CNM)

Input: Objective Function Values, problem dimension

Output: Solutions minimizing objective function values

Begin:
1. Randomly generate a simplex of solutions \( w_i \), \( i = 1, 2, ..., n+1 \), where \( n \) is the dimension of the problem space, and each \( w_i \) is associated with an objective function value \( f_i \) returned.
2. Arrange the solutions according increasing function values such that \( f_i \) represents the \( i \)th best value.
3. Reflect the simplex by Step 2 of NMA and calculate \( w_r \).
4. Subject \( w_r \) to centrifugation followed by truncation \( w_r = \text{Truncate}(\text{Centrifuge}(w_r,t)) \).
5. If \( f_r < f_1 \):
   a. Expand the Simplex by Step 3 of NMA and calculate \( w_c \).
   b. Subject \( w_c \) to truncation and centrifugal force \( w_c = \text{Truncate}(\text{Centrifuge}(w_c,t)) \).
   If \( f_c < f_r \):
      \( w_{n+1} = w_c \) \( f_{n+1} = f_c \)
   Else:
      \( w_{n+1} = w_r \) \( f_{n+1} = f_r \)
End If
6. In case of a failed contraction, shrink the simplex towards the best vertex by Step 5 of NMA.
   Apply centrifugal force to all newly computed vertices \( w_i = \text{Truncate}(\text{Centrifuge}(w_i,t)) \) \( i = 2, 3, n+1 \).
7. Repeat from Step 2 till a convergence criterion is not met.
8. On convergence return the best values of \( w \) (corresponding to minimum objective value).

End

IV. APPLICATION AND CASE STUDY IN FEATURE SELECTION FROM BRAIN NETWORKS

A. Brain Networks

Brain network theory exploits the interconnection between network science, statistics, and neuroscience. It is vehicular in revealing patterns indicative of cognitive states of a person [9], [10]. The brain network theory suggests that the actions performed by the brain are facilitated by interaction among different brain regions. These interacting brain regions are individually called nodes of brain network. Also, the connections among these nodes are referred to as network weights. Over the past few years brain network-based feature extraction techniques have established their significance [24]. Examples of such connectivity measures include correlation [25], coherence [26], Granger causality [27] and phase locking value [28].

B. Brain Network Analysis by Phase Locking Value

A brain network indicates connectivity within brain regions by interconnecting them based on signal similarity obtained from the respective regions. One of the robust ways to calculate signal similarity among brain regions is Phase Locking Value [29], demonstrated as below.

The PLV between two signals how closely their phases are interlocked. A high PLV indicates a consistent phase difference between two signals. The values of PLV range between 0 and 1. It is computed as follows [29], for the EEG signals \( S_x, S_y \).

\[
Z_x = S_x(t) + j\text{HT}(S_x(t))
\]

\[
\Delta \phi_{x,y}(t) = \text{arg} \left( \frac{Z_x(t)Z_y^*(t)}{||Z_x(t)||Z_y^*(t)} \right)
\]

\[
\langle \text{PLV}_{x,y}(t) \rangle = E \left[ e^{j \Delta \phi_{x,y}(t)} \right]
\]

Here \( \text{HT}(\cdot) \) indicates Hilbert Transform.

C. Outline of the Problem

Feature selection is an important problem in many computational environments. A large number of features increases computational time and may also inversely impact the performance of a classification algorithm. In the design of rehabilitative aids, time is of essence, and hence, feature section becomes even more important. For example, for...
rehabilitation of dementia patients [18] it is important to speedily understand their cognitive states and then provide inputs to improve the same. In this module of our work, we attempted to enhance an existing optimization based feature selection technique for recognition of memory encoding versus recall phases of a subject. Feature selection was performed on brain-network features as they were found to be promising [9], [10].

D. Feature Selection by Optimization

The objective function for feature selection [6] was designed, keeping in mind the following criteria

(a) the selected feature set should be such that for different instances of the same class, the values are close. Thus we minimize

\[ J_1 = \sum_{i=1}^{n} |F_{i,k} - \bar{F}_{i,k}| \]

where \( F_{i} \) is \( i^{th} \) feature values of \( k^{th} \) class having \( n \) instances and \( \bar{F}_{i,k} \) is the mean value of the \( i^{th} \) feature of the \( k^{th} \) class.

(b) the selected feature set should maximize difference among the class. Thus we maximize

\[ J_2 = \sum_{i=1}^{n} \left| \frac{\bar{F}_{i,j}}{\sigma_j} - \frac{\bar{F}_{i,k}}{\sigma_k} \right| \]

where \( \bar{F}_{i,k} \) is the mean value of the \( i^{th} \) feature of the \( k^{th} \) class, \( \sigma_j \) and \( \sigma_k \) are standard deviation of the \( i^{th} \) and \( k^{th} \) class. A composite of the above objective criteria is formulated as follows for minimization

\[ J = J_1 + \lambda J_2 \]

Here, \( \lambda \) and is the Lagrange’s multiplier, used to scale the objectives \( J_1 \) and \( J_2 \) to the same magnitude.

E. Stimulus Preparation

The stimulus consist of 5 different objects (diagrams) presented sequentially for 15 seconds each. Each object presentation is followed by a 15 second interval during which a subject is asked recall the object presented immediately earlier. Between each session of encoding and recall of diagrams, a 15 second relaxation interval is allotted. The objects are shown in Fig. 2.

F. Data Acquisition

The experiments were performed on 10 subjects aged 24±5 years. The EEG of the subjects during stimulus presentation was collected from 19 channels placed on the scalp according to the 10-20 electrode system. The channels are (F3, Fz, F4, P4, P3, O1, O2, C3, Cz, C4, F7, F8, T3, T4, T5, T6, Fp1, Fp2, Pz). The signals were windowed to extract relevant instances of encoding and recall. Artifact removal [12] on the signals was performed by thresholding. Then, the signals were filtered at 0-15 Hz, to eliminate the \( \beta \) band, associated with motor activity/planning [8]. Next, feature extraction was performed by computing brain networks by considering phase locking values (PLV)[11] between channels. Sample brain networks

---

Fig. 2: Objects presented for memory encoding and recall
during encoding and recall are shown in Fig. 3. The numbers along the X-axis and the Y-axis represent the EEG signals. These signals are collected from the electrode positions placed according to the 10-20 positioning system [17]. The numbers along the axes represent the positions F3, Fz, F4, P4, O1, O2, C3, Cz, C4, F7, F8, T3, T4, T5, T6, Fp1, Fp2, Pz, in order. The plotted contour map represents connection strengths between the respective scalp areas. The connection strengths are color graded from blue (low PLV) to red (high PLV). The color bar at the right side of Fig. 2 (b) shows the magnitude of phase locking values with their respective color representations. There is a diagonal of red squares in the plotted matrices indicating high PLV of a signal with itself.

G. System Validation

1) Comparison on the basis of feature selection objective values obtained
The proposed algorithm is tested on a well known fitness objective for feature selection [6], and the results are given in Table V. Here, 5 best features are selected among all obtained brain network features. Since the problem space is 5 dimensional, we perform 100 (5×20) function evaluations before declaring the winning algorithm. In Fig. 4, the objective function values obtained (along the Y-axis) are plotted against the number of function evaluations indicated along the X-axis. The plots are those of the Nelder-Mead and Centrifugal NM (CNM), Gravitational Search (GS) Algorithm and Centrifugal GS, and Differential Evolution (DE) and Centrifugal DE (CDE).

![Fig. 4. The objective values obtained by different optimization algorithms with respect to the number of function evaluations (iterations).](image)

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Objective value after 100 iterations</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gravitational Search Algorithm [2]</td>
<td>2.869</td>
</tr>
<tr>
<td>Centrifugal Gravitational Search Algorithm</td>
<td>1.741</td>
</tr>
<tr>
<td>Differential Evolution [1]</td>
<td>2.168</td>
</tr>
<tr>
<td>Centrifugal Differential Evolution</td>
<td>2.121</td>
</tr>
<tr>
<td>Nelder-Mead Algorithm [3]</td>
<td>1.236</td>
</tr>
<tr>
<td><strong>Centrifugal Nelder-Mead Algorithm</strong></td>
<td><strong>1.128</strong></td>
</tr>
</tbody>
</table>

2) Performance of selected features in classification
The features were selected to differentiate between memory encoding and recall. The plausibility of the selected features was tested on a classification problem of distinguishing between memory encoding and recall by employing Support Vector Machines (SVM) [15], [16]. The feature set consists of 171 brain network features extracted from 19 electrodes for each instance. For each of the 10 subjects we performed feature extraction during encoding and recall of 5 different objects. Thus we have a total of 5×10 instances for encoding and the same number of instances for recall. Only 5 features for each instance were selected among all the obtained brain network features during encoding/recall. We compared the (SVM) classification accuracies for Principal Component Analysis (PCA) [13], [14] based EEG feature extraction and the proposed optimization based feature extraction. The results are indicated in Table VI. The classification was performed using leave-one-out cross validation.
TABLE II. THE CLASSIFICATION ACCURACIES OBTAINED USING 10 BRAIN NETWORK FEATURES SELECTED DIFFERENT TECHNIQUES

<table>
<thead>
<tr>
<th>Feature Selection-Classification Algorithm</th>
<th>Classification Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>PCA-SVM</td>
<td>75.82%</td>
</tr>
<tr>
<td>Centrifugal Gravitational Search and SVM</td>
<td>89.97%</td>
</tr>
</tbody>
</table>

V. CONCLUSION

In this work, various optimization algorithms were studied and a common solution to improve the convergence rate of algorithms was proposed. The proposed technique is intuitive and well accepted in other domains of physical systems; it simply involves applying a centrifugal force to increase the rate of sedimentation (here convergence). Centrifugation is performed by adding an exponentially decaying number to the computed solutions and performing non-linear operation on the resultant solution which simulates a circular force. The resultant is truncated within bounds to ensure that infeasible solutions are not generated. It must be also mentioned that the bound constraints of the optimization problem must be known so that we may truncate it and centrifuge it within bounds. The proposed technique was tested on the real world problem of feature extraction for distinguishing between memory encoding and recall. In future, it can be tested on many other problems which require derivative-free optimization.

The real world problem of feature selection for decoding the mental state of a subject was chosen for its applications in neuroscience and rehabilitation. It may also be mentioned that the online systems provide a way to tap into the advantages of the superior temporal resolution of EEG systems. And one fundamental requirement of an online system is a minimal feature set which speeds up mental state recognition. Here, memory related mental state (encoding versus recalling of objects) was studied for its relevance to the dementia related patient community.

The performed experiments have revealed that the optimization based feature extraction technique outperformed the traditional PCA based feature extraction technique. Further, as it is well-known to the computational research community, the applications of optimization are wide and the performance of proposed technique may be studied on multiple real world scenarios related to networking [21], electric power systems [22], design applications[23] and the like.

REFERENCES


