**УДК 004.08**

# ГИБРИДНЫЙ АЛГОРИТМ ВЫБОРА ПРИЗНАКОВ ДЛЯ ВЫСОКОРАЗМЕРНЫХ НАБОРОВ ДАННЫХ НА ОСНОВЕ МЕТОДА РАНДОМИЗИРОВАННЫХ ПРЫЖКОВ, КОТОРЫЙ КОМБИНИРУЕТ ФИЛЬТРУЮЩИЙ И ОБЕРТОЧНЫЙ ПОДХОД

**Брильянтов В.А.** (магистрант, Университет ИТМО)
**Научный руководитель – Сметанников И.Б.**
(доцент ФИТИП, Университет ИТМО)

Главная задача, решаемая в рамках научно-исследовательской работы: реализация алгоритма выбора признаков для высокоразмерных наборов данных на основе метода рандомизированных прыжков, который комбинирует фильтрующий и оберточный методы выбора признаков[1].

**Введение.** В наше время алгоритмы выбора признаков[2] имеют высокую актуальность, поскольку позволяют решать ряд проблем, связанных с размерностью данных. Среди них: переобучение[3] моделей, плохие результаты с точки зрения целевых метрик качества работы моделей, разреженность данных, а также медленная скорость обучения.
Существует два типа алгоритмов понижения размерности – алгоритмы выбора признаков и конструирования признаков. Алгоритмы выбора признаков делятся на несколько типов: фильтрующие алгоритмы, оберточные, ансамблирующие и гибридные.
Основной целью НИР было расширить функционал библиотеки выбора признаков ITMO_FS.

**Основная часть.** IWSSr-SFLA принадлежит к классу генетических алгоритмов отбора признаков. Подобные алгоритмы основаны на идеях создания популяции и её эволюции для поиска оптимальной (с точки зрения какой-либо метрики) особи, представляющей собой решение поставленной задачи. Чаще всего такие алгоритмы используют примеры поведения живых организмов (например, колонии муравьев или стаи волков).

В IWSSr-SFLA используется аналогия с группой лягушек, прыгающих по дискретному пространству в поисках еды (отсюда следует и название алгоритма, Shuffled Frog Leaping Algorithm). Каждая лягушка представляет собой двоичный вектор длины m, где m – число признаков в исходном наборе данных. Единица на позиции i соответствует выбранному i-ому признаку, ноль – его отсутствию, таким образом, каждая лягушка представляет собой набор признаков.

Для создания изначальной популяции используется комбинация алгоритмов IWSSr и Relief Сначала согласно алгоритму Relief определяется важность каждого признака, после чего происходит генерация определенного числа лягушек со случайным числом единиц, причем важность признака пропорциональна вероятности его появления в лягушке. Таким образом, более важные признаки будут появляться в большем числе лягушек.

После этого к такой популяции применяется алгоритм IWSSr, который уменьшает число признаков каждой лягушки. В ходе алгоритма все признаки лягушки сортируются по убыванию метрики Symmetric Uncertainty, после чего формируется новый набор признаков, состоящий изначально из первого по порядку признака:

$$SU_{i,c}(F_i, C) = 2\frac{H(F_i) - H(F_i|C)}{H(F_i) - H(C)}$$

Далее по порядку каждый признак либо замещает собой один из признаков в текущем наборе, либо добавляется к набору, и из всех полученных наборов берется оптимальный с точки зрения работы какого-либо классификатора на входном наборе данных с такими признаками. В результате работы алгоритма уменьшается число признаков в лягушках и повышается их оптимальность.

Для поиска оптимальной лягушки необходимо совершить несколько итераций «прыжков» всей популяции. Оптимальность лягушки, как сказано выше, определяется метрикой работы классификатора на входном наборе данных, в котором остались только содержащиеся в лягушке признаки. На каждой итерации популяция случайным образом делится на заданное количество групп (мемплексов) одинакового размера. Каждый мемплекс лягушек пытается совершить определенное число прыжков в течение одной итерации. Во время каждого прыжка выбирается подгруппа (субмемплекс) лягушек (одинакового размера во всех мемплексах), причем вероятность попадания лягушки в субмемплекс пропорциональна ее оптимальности. После выбора субмемплекса худшая лягушка в нем пытается совершить «прыжок» в сторону лучшей лягушки субмемплекса. Прыжок происходит путем добавления к худшей лягушке случайного числа признаков, которые имеются у лучшей лягушки и отсутствуют у худшей, либо удаления случайного числа признаков, которые имеются у худшей лягушки и отсутствуют у лучшей:

$$S_b = \begin{cases} \min\{\text{int}(\text{rand}[SP_b - SP_w]), S_{max}\} & \text{if } SP_b > SP_w \\ \max\{\text{int}(\text{rand}[SP_b - SP_w]), -S_{max}\} & \text{else} \end{cases}$$

После совершения прыжка проверяется оптимальность лягушки – если она увеличилась, то новая лягушка замещает старую; в противном случае лягушка пытается аналогичным образом совершить прыжок (со старым набором признаков) в сторону лучшей лягушки популяции. Если и такой прыжок не улучшил оптимальность лягушки, производится попытка случайно сгенерировать лягушку и заменить ей худшую лягушку в случае увеличения оптимальности.

Input: Training Data
Output: S: The selected subset
Parameters: N: number of features /Par: Parameters of SFLA /
/ m: Repetition of the whole process of Relief method

////////////// **Filtering Section** ////////////////

Set W[J] =0
For i=1:m
  Randomly select an instance R
  Find nearest hit H and nearest miss M
  For J =1 to N do
    W[J] = W[J] – diff(J,R,H)/m + diff(J,R,M)/m
Probsel[i]= W [i]/$\sum_{j=1}^{N}$ W[j]

////////////// **IWSSr_SFLA Section** ////////////////

Generate the initial population by using Probsel
Apply IWSSr Algorithm
Evaluate the initial population using Fitness function
while (Itr < $IT_{max}$)
  Partition the population into sfla_m memplex
  k=0
  while( k < sfla_m)
    k=k+1
    select k-th memeplex;
    i=1
    while (i < $IT_{mem}$)
      Generate a submemeplex base on $P_j = \frac{2(sfla\_n+1-j)}{sfla\_n\ (sfla\_n+1)}$ $.j = 1.2. \ldots . sfla\_n$

Leaping

      Select the best frog $F_b$ from memeplex
      Select the worst frog $F_w$ from submemeplex
      $P'_w$ =IWF($F_G$, $F_w$)
      $fit$ =Evaluate($P'_w$)
      If ($fit(P'_w) > fit(F_w)$)
        Replace $F_w$ with $P'_w$
        i=i+1
      else
        Select the best frog $F_G$ from whole population
        $P''_w$ =IWF($F_G$, $F_w$)
        $fit$ =Evaluate($P''_w$)
        If ($fit(P''_w) > fit(F_w)$)
          Replace $F_w$ with $P''_w$
          i=i+1
        else
          Randomly generate a new frog ($P'''_w$)
          $fit$ =Evaluate($P'''_w$)
          If ($fit(P'''_w) > fit(F_w)$)
            Replace $F_w$ with $P'''_w$
            i=i+1
          else
            $F_w = F_w$
  Shuffle all the frog
  Itr= Itr +1

S= $F_G$

**Figure 5.** Pseudo code of the proposed hybrid algorithm.

Таким образом, в пределах каждой итерации лягушки с более низкой оптимальностью совершают прыжки в сторону лягушек с более высокой оптимальностью, и вся популяция лягушек движется к оптимальному набору признаков. Перемешивание мемплексов после каждой итерации способствует выходу лягушек из возможных локальных максимумов. Также в некоторых реализациях алгоритма присутствуют генетические операторы, например мутация или кроссовер, применяемые после каждой итерации. Итогом работы алгоритма становится лучшая лягушка популяции, которая описывает собой оптимальный набор признаков.

**Выводы.** В результате выполнения НИР был имплементирован гибридный алгоритм выбора признаков, что позволило увеличить количество задач, решаемых с помощью библиотеки.