Extension of Direct Search Methods to find Optimal Cluster Centroid for Constrained Multi-Variable Functions

Reddy K. Madhavi¹, Vinaya A. Babu², Ananda A. Rao³ and Viswanadha S. Raju⁴
¹Research Scholar, JNTUA, Anantapur, India
kreddyadhav@gmail.com
²Professor of CSE, JNTUH, Hyderabad, India
{avb1222, svraju.jntu}@gmail.com
³Professor of CSE, JNTUA, Anantapur, India
akepogu@gmail.com

Abstract— Identification of useful clusters in large datasets has attracted considerable interest in clustering process. Since data in the World Wide Web is increasing exponentially that affects on clustering accuracy and decision making, change in the concept between every cluster occurs named concept drift. The new data must be assigned to any one of generated clusters called data labeling. To say that data labeling was performed well the clusters must be efficient. Selecting initial cluster center (centroid) is the key factor that has high affection in generating effective clusters. So we are concentrated in proposing methods to identify initial cluster center based on the properties of data. Our previous work was concentrated on selecting optimal initial cluster centroid for unconstrained functions by applying direct search methods. To reduce the difficulties arised during application of these direct search methods to constrained functions, this paper proposes a method to select initial cluster centroid for constrained multi-variable functions then apply any existing clustering algorithm.

Index Terms— Concept drift; Sliding window; cluster centroid and constrained multi-variable functions.

I. INTRODUCTION

Data mining refers to extracting or mining knowledge from large volumes of data [3]. One of the most distinct characteristics of data mining is that it deals with very large and complex datasets (gigabytes or even terabytes) often contain millions of objects described by tens, hundreds or even thousands of various types of attributes or variables (interval, ratio, binary, ordinal, nominal etc.). This requires the data mining operations and algorithms to be scalable and capable of dealing with different types of attributes. Clustering is one of the major functionalities and is often a critical component of the data mining or information retrieval process.

Clustering is an unsupervised learning and clusters are generated based on the principle of low-interclass similarity and high-intraclass similarity [3]. Data has been clustered around centroids. The existing clustering algorithms randomly selects any data point as initial cluster center and performs the clustering process. But different random initializations of cluster centers [4] can lead to different final number of clusters. Therefore choosing the suitable initial cluster centers is a problem worthy of studying.

The rest of the paper is organized as follows: section 2 concentrates on relevant study, section 3 provides methods for finding initial cluster centroid for multivariable constrained functions, section 4 presents illustration of the proposed method and section 5 concludes with summary of the paper and its future work.

II. RELEVANT STUDY

At each time, new data enters into the database. To add an importance of this new data to the clusters, that data can be handled properly. So, basically, we need a way to identify those elements of the stream in a timely manner that is no longer consistent with the current concepts. Instead of sampling the data stream randomly, we can use the sliding window model to analyze stream data. The basic idea is that rather than running computations on all of the data seen so far, or on some sample, we can make decisions based only on recent data like in stocks or sensor networks. More formally, at every time t, a new data element arrives. This element “expires” at time t+z, where z is the window size or length. It also reduces memory requirements because only a small window of data is stored. The intuition behind it is to incorporate new examples yet eliminate the effects of old ones. We can repeatedly apply a traditional classifier to the examples in the sliding window. As new examples arrive, they are inserted into the beginning of the window. The corresponding numbers of examples are removed from the end of the window and the classifier is reapplied. This technique, however, is sensitive to the window size, z. If z is too large, the model will not accurately represent the concept drift. Drift means change in concept with respect to time. On the other hand, if z is too small, then there will not be enough examples to construct an accurate model.

The incoming data points in the sliding window [5] should be able to be allocated into the corresponding proper cluster at the last clustering result. Such process of allocating the data points to the proper cluster is named as “data labeling” [2]. After obtaining temporal clustering results by data labeling, these clusters are compared with the last clustering...
results, which are base for the formation of the new clusters. For the proper data labeling process, the clusters must be efficient. In the process of generation of these efficient clusters, selecting the initial cluster center has high impact. In existing methods the initial cluster center was selected randomly. A survey on selection of initial cluster center in various methods were presented in [4]. Different initializations results in different clustering results. That is number of data points in a cluster is sensitive to selection on initial cluster center. So instead of random selection we are proposing methods to identify initial cluster center by performing preprocessing that is to analyze the properties of data. Depending on these properties the objective function is formulated [20,26]. Our previous work was provided for selecting initial cluster center for single variable functions, two variable functions and direct search methods for unconstrained multi variable functions. For the constrained functions, to accept any point the constraints could be checked. If that point is violated then that current point is rejected and another point that would satisfy constraints is generated. If we acclimatize our direct search methods like simplex search, pattern search and conjugate direction methods [6,7,8] to constrained functions, the problems in view of line search, rate of convergence, feasibility and termination etc occurs. The use of any set of search directions that does not adjust itself to allow movement along constraint surfaces is fairly unsatisfactory. This can be alleviated if search directions are widely distributed. This is achieved [1] with the following proposed method to accommodate inequality constraints and assumes the elimination of equality constraints.

III. PROPOSED METHOD

This method is an extension of direct simplex search method and most widely used in many engineering applications for nonlinear unconstrained multivariable functions. For given initial strictly feasible point \( y^0 \), reflection parameter \( \alpha \) and termination parameters \( \beta \) and \( \gamma \).

1. Generate the initial set of \( F \) feasible points and for each point \( f=1,2,\ldots,F-1 \)
   a. Determine the point \( y_f^i \) by sampling \( S \) times.
   b. If \( y^i \) is infeasible, calculate the mean \( \bar{y} \) of current points and reset
      \[ y^i = 0.5(y^i + \bar{y}) \]
      Now repeat this computation till \( y^i \) becomes feasible. If it is feasible then sample \( S \) times until we find \( F \) points and compute \( f(y^i) \) for \( f=1,2,\ldots,F-1 \)

2. Reflection step
   a. Take point \( y^k \) such that \( f(y^k) = \max f(y^i) = \text{MAX} \)
   b. Find the mean \( \bar{y} \) and new point
      \[ y^o = y^k + \alpha (y^k - \bar{y}) \], here \( \alpha > 1 \)
   c. If \( y^o \) is feasible and \( f(y^o) \) e” \( \text{MAX} \), retract half the distance to the centroid . Continue until \( f(y^o) < \text{MAX} \).
   d. If \( y^o \) is feasible and \( f(y^o) < \text{MAX} \), then go to step 4.
   e. If \( y^o \) is infeasible then go to step 3.

3. Adjust for feasibility
   a. For violated variable extremes, reset
      If \( y_{i}^+ < y_{i}^- \), then set \( y_{i}^+ = y_{i}^- \)
      If \( y_{i}^+ > y_{i}^- \), then set \( y_{i}^- = y_{i}^+ \)
   b. If the resulting \( y^o \) is infeasible, retract half the distance to the centroid. Continue until \( y^o \) is feasible, then go to step 2(c).

4. Termination
   \[ \text{Find} \quad \bar{f}(y) = \frac{1}{F}\sum f(y_f) \quad \text{and} \quad \bar{f} = \frac{1}{F}\sum f^2 \]
   If \( \sum[f(y^f) - \bar{f}]^2 \leq \beta \) and \( \sum|f(y^f) - \bar{f}| \leq \gamma \)
   Terminate, otherwise go to step 2(a).

IV. ILLUSTRATION

Let us introduce the variables \( y_1 \) is depth, \( y_2 \) is width, \( y_3 \) is height to design rectangular structure with an open frontal face to minimize material cost with volume of 16,000 ft\(^3\), perimeter should be no more than 220 ft, depth is to be no more than 60 ft and the width no more than 80 ft. The width should not exceed three times the depth, and the height is to be no more than two-thirds of the width. The cost of the corrugated material of which the three sides and roof are to be constructed is $30/ft\(^2\). The material cost will be given by cost of roof \( 30y_2 \), cost of backwall \( 30y_2y_3 \) and the cost of sides \( 2(30y_2y_3) \).

The objective function for this problem can be formulated by eliminating the equality constraint \( y_3y_1 = 16000 \) with the proper substitution as \( y_3 = 16000/y_1y_2 \).

The objective function with the constraints becomes

Minimize \( f(y) = 30y_1y_2 + 480000/y_2 + 960000/y_1 \)

Subjected to constraints

1) \( y_2 - y_3 \geq 0 \), \( 3y_2 - y_3 \geq 0 \), \( 0 \leq y_3 \leq 30 \), \( 0 \leq y_2 \leq 70 \).

Find out feasible region for this problem. Procedure to find out feasible region was illustrated in one of our previous papers.

Let us initiate this method by taking the initial feasible point \( y^{(0)} = (42, 42) \). Three additional points are generated because our problem is a two-dimensional problem. Each point is generated as \( y^{(2)} = (0 + (r_1 \times 60), 0 + (r_2 \times 70)) \). Here \( r_1 \) and \( r_2 \) are random numbers generated in the interval \((0,1)\). Let \( r_1 = 0.4, r_2 = 0.8 \). Now \( y^{(2)} = (24, 56) \). This point is in the feasible region.

Let random numbers generated to find \( y^{(3)} \) are \( r_1 = 0.2, r_2 = 0.3 \). Therefore \( y^{(3)} = (12, 21) \), this is in the infeasible region. To make this point feasible, retract this point towards the centroid of remaining two feasible vertices \( y^{(1)} \) and \( y^{(2)} \).

\[ \bar{y} = 0.5(y^{(1)} + y^{(2)}) = (32, 48) \] Then \( y^{(3)} = 0.5(y^{(3)} + \bar{y}) = (22, 34.5) \). It is feasible. Let the random numbers generated to find \( y^{(4)} \) are \( r_1 = 0.8, r_2 = 0.4 \). Then \( y^{(4)} = (48, 28) \). It is feasible. Now we are ready with four points.
\( f(y(1)) = 30 \times 42 \times 42 + \frac{480000}{42} + \frac{960000}{42} = 87205.714 \). Similarly \( f(y(2)) = 77462.85 \), \( f(y(3)) = 72414.26 \) and \( f(y(4)) = 84605.71 \). Now maximum function value is at \( y(1) \). Hence \( y(1) \) is the point to be reflected out called worst vertex. New candidate point is obtained with step size \( \alpha = 1.01 \). Compute \( \bar{y} \) as the mean of three points other than \( y(1) \), that is mean of \( y(2), y(3) \) and \( y(4) \). Therefore \( \bar{y} = (31.34, 39.5) \), it is feasible. Now \( \bar{y} = \bar{y} + \alpha(y(1) - \bar{y}) = (20.58, 37) \). \( f(y(5)) = 72113.36 \), which is less than \( f(y(1)) \). Now current set of vertices consists of \( y(2), y(3), y(4) \) and \( y(5) \). Now the worst vertex among these vertices is \( y(4) \) which is the point that is to be reflected out. Now the process is repeated to get the new vertex. Repeat the iterations until we get the optimum. Optimum solution is the solution at which we get minor difference in the function value from one iteration to next iteration.

V. Conclusion And Future Work

This paper proposes the method to identify optimal cluster centroid for constrained multi-variable functions that is simple. The optimum point that is obtained at the end of illustration is the minimum point which intuit act as initial cluster centroid for the generation of clusters. This method suitably works well than the direct search methods that are applied to constrained multi-variable functions. This method can be further extended to apply on popularly used datasets.

ACKNOWLEDGEMENT

Our sincere gratitude to the professors of JNTU for their valuable suggestions for the preparation of this paper.

REFERENCES