Transfer Learning for Artificial Bee Colony Algorithm to Optimize Numerical Functions

M. HACIBEYOGLU, B. KOÇER, and A. ARSLAN

Abstract—Most of the traditional machine learning techniques are based on training a learner system with training data and making prediction over test data which has the same distribution with training data, but sometimes this rule breaks and distribution of test data may differ from training data. In such a case instead of training a new model from scratch, transferring feasible knowledge from old model can reduce training time and improve predictive accuracy. At this point transfer learning techniques can apply to transfer knowledge between tasks. In this work we developed a transfer learning method that can be used with artificial bee colony algorithm to optimize numerical functions.

Keywords—Artificial bee colony algorithm, transfer learning, swarm intelligence.

I. INTRODUCTION

TRAIN and test is the main principle of supervised machine learning methods. This principle can be summarized as training a learner with available training data and testing similarly distributed test data. However sometimes in real life this principle doesn’t work. Distribution of test and training data may be differ in time or trained model may have to be adapted to a similar task. Even sometimes it may be very hard to find labeled training data. In such cases transfer learning methods can be used. Transfer learning is transferring knowledge from related source tasks to target task in order to improve performance of target task in insufficient or absence of labeled training data. The knowledge which transferred between tasks may be weighted instances of source task data, a common feature representation that reduces differences between source and target task or shared parameters. Domain adaptation [2], covariate shift [3] and sample selection bias are also related research areas. Figure 1 illustrates differences between traditional machine learning and transfer learning methods.

II. RELATED WORK

There are a lot of naturally inspired algorithms. Most used one is genetic algorithms (GA). However GA cannot guarantee global optimum solutions due to population diversity [5], swarm intelligence has attracted more interest in recent years. For example ant colony optimization [6] which simulates ants seeking a path to find closest path between food source and their colony and particle swarm optimization [7] which was born from the research of “how can birds or fish exhibit such a
coordinated collective behavior?”. Bee colony optimization [8] is another swarm intelligence based algorithm like artificial bee colony algorithm [1]. Both of the studies are inspired by natural behaviors of bees while looking for food. Artificial bee colony algorithm is used to train neural network [9] and to medical pattern classification [10] and clustering problems [21].

Like genetic algorithms and swarm intelligence algorithms, transfer learning methods can be categorized biologically inspired methods too because it is an attempt to simulate decision system of human intelligence using past experiences. Transfer learning methods are applied many real world problems some of them indoor wi-fi localization task [11,13] which collects data to build a model that predicts the position of receiver in an office and transfers the knowledge for different model when model is inaccuracy due hardware change or over time, Chinese web page classification using English web pages [12] to overcome information bottleneck in Chinese web page classification task, lifelong robot learning [14] which reduces real world experiments amount and training time in new scenarios and even image classification [15] which utilizes priors from a related categories and unlabeled data for learning new visual category. Some of traditional machine learning methods are modified for using transfer learning like genetic algorithms [16] to improve performance of similar numeric functions solutions, markov logic networks [17] to transfer and revise relationships from one domain to another and reinforcement learning for skill [18], action schemas [19] and control knowledge [20] transfer.

III. ARTIFICIAL BEE COLONY ALGORITHM

Aim of the ABC Algorithm is getting maximum available food with minimum energy spend. Algorithm works in four phases and uses three type of bee.

Phase 1- Initialization phase: First phase is randomly creating food resources.

Phase 2- Employee bee phase: Each food resource is assigned to an employee bee and better food resources are searched.

Phase 3- Onlooker bee phase: Onlooker bees search better food resources around the current resources. Better resources with high fitness values attract more onlooker bee.

Phase 4- Scout bee phase: Food resources whose fitness can’t be increased anymore by onlooker bees are left and new food resources are searched by scout bees.

IV. TRANSFER LEARNING FOR ARTIFICIAL BEE COLONY

Artificial bee colony algorithm visits a lot of point while exploration and exploitation processes. It exploits the search space via scout bees and explores food resources neighbor via onlooker and employee bees. Like other biologically inspired algorithms artificial bee colony is an iterative algorithm and in every new iteration algorithm forgets previously visited points in the search space expect food resources but food resources are also subject to change. Origin of our method is that the forgotten points may be useful for similar tasks. In a classification task, similar tasks can be described easily. For example in a document classification task, documents from different sub categories can be used to knowledge transfer between themselves to improve predictive accuracy in absence of labeled data. In optimization, similar tasks can be described having near optimal solution and similar fitness evaluation functions.

Fig. 2 Modified ABC algorithm for transfer learning. Added parts for transfer learning are shown with italic and red text.

Proposed method is a modified version of artificial bee colony algorithm which is illustrated in figure 2. Added parts of the proposed method in both source and target tasks are showed with italic and by red text. In source task three food source positions are stored in each iteration in order to evaluate for target task and transfer best suitable ones. These points are determined by trial counter of food sources. For this purpose, food sources are sorted by trial counts and position of three food sources which have the biggest, middle and the smallest trial counts are stored. When source task finished, stored points are evaluated for target task and 25% of the biggest fitness valued positions are assigned to randomly
created food resources and afterward ABC algorithm goes on as usual.

V. EXPERIMENTAL RESULTS

We used same shifted functions which have different optimum solutions. So similarities between source and target task can be described by distance between optimal solutions. For example when shifted sphere functions (1) is regarded as source task and (2) is regarded as target task, similarities between source and target task can be described by ratio of Euclidian distance between optimal solutions and range of variables (9). In equation 9 “lb” represent lower bound and “ub” represent upper bound of variable “x” For computation simplicity we used f bias=0 for all functions.

\[ S = \frac{1}{ub-lb} \times \sqrt{\sum_{i=1}^{D} (a_i - b_i)^2} \]  

Variables which is calculated from equation 9 represents the similarity of source and target task. Lower S values represent small distance between the solutions of the target and source task i.e. more similar tasks and higher S values represents more different tasks. 50 dimensional functions (D=50) are used, limit value is set to 2500 and population size is set to 100 for both source and target tasks in all tests. We used mean of 30 independent runs to draw the graphics from figure 3 to 6. Graphics represents convergence results for different S values.

![Fig. 3 Convergence graphic for different S values. Source task is F1 and target task is F2.](image3)

![Fig. 4 Convergence graphic for different S values. Source task is F3 and target task is F4.](image4)

<table>
<thead>
<tr>
<th>No</th>
<th>Function</th>
<th>Formulation</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Shifted Sphere</td>
<td>( F_1(x) = \sum_{i=1}^{D} z_i^2 + f_{bias} ) ( a = [a_1, ..., a_D] ) ( x = [x_1, ..., x_D] )</td>
<td>( x \in [-100,100] )</td>
</tr>
<tr>
<td>2</td>
<td>Shifted Sphere</td>
<td>( F_2(x) = \sum_{i=1}^{D} z_i^2 + f_{bias} ) ( b = [b_1, ..., b_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
<tr>
<td>3</td>
<td>Shifted Schwefel</td>
<td>( F_3 = \sum_{i=1}^{D} \sum_{j=1}^{D} z_i^2 + f_{bias} ) ( a = [a_1, ..., a_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
<tr>
<td>4</td>
<td>Shifted Schwefel</td>
<td>( F_4 = \sum_{i=1}^{D} \sum_{j=1}^{D} z_i^2 + f_{bias} ) ( b = [b_1, ..., b_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
<tr>
<td>5</td>
<td>Shifted Rosenbrock</td>
<td>( F_5 = \sum_{i=1}^{D} \frac{100(z_i^2 - z_{i+1})^2}{(z_i - 1)^2} + f_{bias} ) ( a = [a_1, ..., a_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
<tr>
<td>6</td>
<td>Shifted Rosenbrock</td>
<td>( F_6 = \sum_{i=1}^{D} \frac{100(z_i^2 - z_{i+1})^2}{(z_i - 1)^2} + f_{bias} ) ( b = [b_1, ..., b_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
<tr>
<td>7</td>
<td>Shifted Rastrigin</td>
<td>( F_7 = \sum_{i=1}^{D} 10 \cos(2\pi z_i) + 10 + f_{bias} ) ( a = [a_1, ..., a_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
<tr>
<td>8</td>
<td>Shifted Rastrigin</td>
<td>( F_8 = \sum_{i=1}^{D} 10 \cos(2\pi z_i) + 10 + f_{bias} ) ( b = [b_1, ..., b_D] ) ( x \in [-100,100] )</td>
<td>( x = [x_1, ..., x_D] )</td>
</tr>
</tbody>
</table>
VI. CONCLUSION

We used transfer learning for artificial bee colony algorithm. Experimental results have showed that knowledge transfer is possible between two numerical optimization problems for ABC algorithm and when optimal solutions of source and target functions get closer performance of target task increases. For future work we will try knowledge transfer for classification tasks via artificial bee colony algorithm and improve transfer performance.

REFERENCES


Web Representation of Spatial Data

Kapil Aggarwal

Abstract—In spatial applications, it is often necessary to use spatial objects with varying degree of detail. As spatial objects are typically displayed in a way that allows human users not to discern unnecessary details, it would suffice to draw an abstraction of the spatial data preserving its characteristics. Generalization, the process to derive such a less detailed representation, may lead to a remarkable reduction of the computational overhead involved with displaying complex spatial objects. It can also benefit data transfer from one site to another. With a growing number of interactive spatial applications on the web, spatial data generalization becomes increasingly important. In this paper, we investigate how database systems can support generalization of spatial data. We introduce the concept of visual significance of spatial objects for interactive spatial applications, and suggest efficient ways to produce a good quality generalized map for web-based spatial applications.

Keywords—Generalization, Spatial indexing, Spatial object, Visual significance.

I. INTRODUCTION

Spatial database systems (SDBS) and Geographic Information Systems as their most important application aim at storing, retrieving, manipulating, querying, and analyzing geometric data [1]. During the last two decades, spatial database systems have become a vital part of Geographical Information Systems (GIS) for storing and accessing spatial data. Spatial Databases is the first unified, in-depth treatment of special techniques for dealing with spatial data, particularly in the field of geographic information systems (GIS). Spatial database systems offer the underlying database technology for geographic information systems and other applications. A spatial database is a specialized type of relational database which is optimized to store and query geographic data, including points, lines and polygons. While typical databases can understand various numeric and character types of data, additional functionality needs to be added for databases to process spatial data types such as shapes. Their role is pronounced by the steadily increasing amount of spatial data maintained in such systems. While the data stored in a database represents the finest level of details available, for a given application it is often desirable to use a level of details suitable for the application. The derivation of an abstract representation is known as generalization. Generalization derives from a source dataset a target dataset at a reduced scale whose contents and complexity have been reduced in such a way that the structural characteristics of the source data are maintained for a given application. By removing excessive and non-relevant details, it is possible to derive a representation of the source data that is much more suitable for the given application scenario. Generalization can reduce the data volume considerably since a target data set typically consists of fewer and simpler data objects than the source data. This performance aspect of generalization becomes increasingly important with interactive spatial applications using spatial vector data on the Internet. Over past two decades, tremendous research efforts have been spent in GIS community on the knowledge acquisition for automatic generalization based on the rule-based systems [2].

A general architecture for Internet-based spatial applications uses a spatial database to store both spatial and non-spatial data. The database can be browsed by means of a Web-browser capable of running Java applets. Generating requests and displaying the result data is handled by the applet that runs on the user's machine. On the server site, a database web server translates the user's request into queries against the spatial database to fetch qualifying objects. These objects are then encoded according to some spatial data transfer protocol and transferred to the applet over the network. After decoding the data on the client side, they are drawn on the screen. Generalization can improve the performance on both the database side & the application side. Obviously, once the objects are retrieved from the database, they can be simplified on the server site by removing parts of an individual spatial object or finding a suitable abstraction that preserves its characteristics. This type of simplification is called as object generalization. In this paper, more focus is given on a specific type of generalization, called as data set generalization. It deals with removing objects from data sets and addresses the issue of spatially significant objects. It approaches the problem of generalization from a database perspective, treating generalization as an integral part of database query processing. A database query generated from a user's request is modified taking into account of how these data are to be used. If some objects cannot be perceived when being displayed in the web browser, data set generalization tries not to retrieve them from the database at all. However, generalization can degrade the quality of map if the characteristics of the original map are not preserved during generalization. In addition, the data set generalization step may also increase the overall processing.