Feature weighting using particle swarm optimization for learning vector quantization classifier

A Dongoran*, S Rahmadani†, M Zarlis† and Zakarias†

†Magister Teknik Informatika, Fasilkom - TI, Universitas Sumatera Utara (USU), Medan - Indonesia

*Awaluddin.dongoran@gmail.com

Abstract. This paper discusses and proposes a method of feature weighting in classification assignments on competitive learning artificial neural network LVQ. The weighting feature method is the search for the weight of an attribute using the PSO so as to give effect to the resulting output. This method is then applied to the LVQ-Classifier and tested on the 3 datasets obtained from the UCI Machine Learning repository. Then an accuracy analysis will be generated by two approaches. The first approach using LVQ1, referred to as LVQ-Classifier and the second approach referred to as PSOFW-LVQ, is a proposed model. The result shows that the PSO algorithm is capable of finding attribute weights that increase LVQ-classifier accuracy.

1. Introduction
Classification is a process of learning by mapping a data into one or more predefined classes based on information discribed by its own features [6]. Learning that occurs in the classification process is also called an inductive-learning algorithm [1]. The purpose of the classification is to create a model that will be used to predict the value of the input data that is not yet known class. Simply split the classification process usually divide the sample data into two parts [2]. First referred to as training data used to train the data to get the expected classification model. The two are referred to as test data. This test data will be tested to determine the level of accuracy in predicting the value of new input data, or can also known error rate predictions.

A statistical approach can also be used to generate a classification model. In addition to statistical approaches, there are many more approaches used to generate classification models, such as decision tree, artificial neural network, genetic algorithm, evolutionary computing, swarm optimization, etc.

In this paper we will use competitive learning method based on competition on LVQ competitive network, and using particle swarm optimization (PSO) method to weighting each attribute of sample data. Both of these approaches will be compared with each other to find out the performance of both approaches in predicting the value of new mapped input data. The weighting method feature improves performance in this case increases accuracy or decreases error rate [5].

The weighting feature is part of the selection feature. The selection feature is a fairly complex job that can affect the performance of classification. Therefore approaches to feature selection continue to be developed. The most popular current approach is based on Evolutionary Computation (EC), Particle Swarm Optimization (PSO) [7] [8] based Swarm Intelligence. PSO is a simple and fast
computing technique that produces convergent values. So PSO is an effective technique used for feature selection tasks [10].

2. Learning Vector Quantization (LVQ)

Learning vector quantization (LVQ) is a method that can be used to generate a classification model of trained sample data. LVQ is part of a supervised neural network method [9]. LVQ architecture is similar to Kohonen's SOM (Self Organizing Map) developed by Prof. Teuvo Kohonen in 1982.

Simply put, competitive layer on artificial neural networks have a single output layer, and each neuron is fully connected to the input node [1].

![Architecture Simple LVQ](image)

**Figure 1.** Architecture Simple LVQ

2.1. LVQ Algorithm

The value of each attribute (information) training data is used to form a weight vector (prototype). The weight vector then iteratively becomes a model that will be used to classify the new sample data whose class is not yet known. The first stage of the LVQ algorithm is to determine the initial weights in the classification model search process. The initial weights used are derived from the training data samples. LVQ applies a winner-take-all strategy [5], choosing a winning vector as a winner. The winning weight vector is the vector that has the smallest distance or in other words, this weight has a resemblance to the input vector. Then the winning weights vector are updated or adjusted. While other weight vectors (which are not winners) are not updated. The weight vector update process is performed repeatedly until the termination conditions are met. The size of the similarity between the weight vector and the input vector is determined using the distance (Euclidean) formula in equation 1 below.

\[
dist(x, w) = \sqrt{\sum_{i=1}^{n}(x_i - w_i)^2}
\]  

Where \( n \) is the number of attributes, \( x \) is the input vector, and \( w \) is the weight vector.

In pattern recognition, the minimum Euclidean distance between the input vectors and the vector weights is the winner. This winning vector is then updated. If the value of input vector \( x \) and weight vector \( w \) (winner) are in the same class, then the weight vector \( w \) is updated using equation 2.

\[
w(t + 1) = w(t) + \alpha (x(t) - w(t))
\]  

And vice versa, if the input vector \( x \) and weight vector \( w \) (winner) are in different classes, and then the weight vector \( w \) is updated using equation 3. Where \( \alpha \) is called the learning rate.
\[ w(t + 1) = w(t) - \alpha (x(t) - w(t)) \]  

The steps of LVQ learning process to generate the classification model, as follows:

1. Input: Input data, vector codebook, learning rate, maximum iteration
2. Initialization vector weights (taken from data input)
3. For each data input, calculate the Euclidean distance between each data input \( x \) and the weight vector \( w \) using equation 1
4. For each weighted vector \( w \), select the smallest Euclidean distance as the winner.
5. Update the winning weight vector \( w \):
   - If class \( w \) is equal to class \( x \), update \( w \) using equation 2.
   - If class \( w \) is not the same as class \( x \), update \( w \) uses equation 3.
6. If the termination condition is met, print the weighted vector \( w \) as the resulting classification model, otherwise return to step 4

3. Particle Swarm Optimization (PSO)

The swarm particle is a population-based optimization technique developed by James Kennedy and Russ Eberhart in 1995 [7] [8] inspired by the behavior of bird flocks. Although Kennedy and Eberhart started modeling with flocks, it turned out that in its development, the algorithm model is more similar to a model of swarming bees with collective intelligence [3].

3.1. PSO Algorithm

The particle swarm-based optimization begins by initializing the positions of the particles randomly. Then each particle calculated its fitness value based on the evaluation function. This method will continue to iterate until the expected condition is reached (optimal condition). At each iteration, each particle is affected by two values [5]. The first is the best value each particle has achieved, called the \( pbest \). The other value is the best overall value possessed by particles in the population, referred to as \( gbest \). After obtaining the two best values, the particles renewed the velocity and their respective positions with equations 4 and 5.

\[
v_i(t) = v_i(t-1) + c_1 r_1 (x_{pi} - x_i) + c_2 r_2 (x_g - x_i) \]  
\[ x_i(t) = x_i(t-1) + v_i(t) \]  

Where \( v \) represents velocity, \( x \) is the particle position, \( c \) is the learning rate and \( r \) is a random number \( U(0,1) \).

\( v_i(t - 1) \) is also called inertia term [11] that forces particles to move in the same direction as the previous movement. \( c_1 r_1 (x_{pi} - x_i) \) Is referred to as the cognitive term [11] that forces the particle
back to its best position. \( c_2 r_2 (x_g - x_i) \) is referred to as social term [11] which forces particles to move to the best position of all particles.

The steps for the particle swarm-based optimization procedure are as follows:

1. Generate the initial population of the particles
2. Calculate the fitness value, if better than the fitness value during the run, set that value as \( pbest \)
3. Choose the particle with the best fitness value in the population, set that value as \( gbest \)
4. For each particles:
   a. Update velocity \( v \) using equation 4
   b. Update position \( x \) using equation 5
5. Stop if expected conditions are reached. If not, returned to step 2

4. Experiment

In this paper used two methods, namely LVQ and PSO. PSO as a weight seeker on feature (attribute) sample data, while LVQ is used to find the classification model (classifier). In the PSO algorithm, the data input feature is used to search for the weight value, then the weight value is used as the LVQ input parameter to generate the classification model through the training process. The classification model is then validated in the testing process. The performance level is then calculated for looking at the accuracy and error rate. The data used for training and testing is obtained from sample dataset using Rotation (n-fold cross validation).

4.1. Model PSO to find Feature Weighting

The particle swarm-based optimization begins with the initialization of the population of the particles by randomize way. The particles represent the sample dataset’s attributes. The value of each attribute is randomized (from the minimum to the maximum value) to set the initial position of the particle. While the initial velocity of the particles is set to zero. The particle density must not exceed the maximum value of each attribute of the sample dataset. After initializing the value, each particle is then evaluated by an evaluation function to find its \( pbest \) value [5]. From the value obtained \( pbest \), select one value with the best value as the \( gbest \) value [5]. Then update the velocity and position of each particle using equations 4 and 5. The fitness value evaluation process, density and position update, is done iteratively until the expected termination conditions are achieved. The value of the learning rate is updated iteratively to see the resulting convergence level. Convergence is met if the evaluation function tends to improve and stagnate, and the position of the particles has not changed significantly. The output of this PSO algorithm is the weighted value of each attribute of dataset.

4.2. Model LVQ to find Model Classifier

The weight of the attributes obtained from the PSO algorithm is then applied to the LVQ model. The weight of the attribute \( a_j \), where \( j \) is the \( j \)’s attribute, is inserted into the euclidean distance formula, as shown in equation 6.

\[
dist(x, w) = \sqrt{\sum_{i=1}^{n} (a_j * (x_i - w_k))^2} \tag{6}
\]

Where \( x \) is the input data, \( w_k \) is vector weight for each class \( k \), \( a_j \) is the feature weight generated by the PSO on each attribute \( j \).

While the formula for the winner vector weighted update is modified due to the weight value of the resulting attribute. If the input \( x \) and the weight vector \( w \) are in the same class then the weight \( w \) is updated using equation 7, otherwise if \( x \) and \( w \) are not in the same class, the weight \( w \) is updated using equation 8.
\[ w_{(t+1)} = a \ast w_{(t)} + \alpha \left( a \ast (x_{(t)} - w_{(t)}) \right) \] (7)  
\[ w_{(t+1)} = a \ast w_{(t)} + \alpha \left( a \ast (x_{(t)} - w_{(t)}) \right) \] (8)

Where \( a \) is the attribute’s weight, \( \alpha \) is a learning rate.

5. Result
The dataset that used to train and test the above experiments was obtained from UCI Machine Learning Repository. The dataset has been classified before, each instance already has a certain class label. Table 1 shows a description of the 3 datasets with property A representing an attribute, \( I \) denoting the number of instances, \( C \) denoting the number of classes. Table 2 shows the optimal attribute weights generated by the PSO algorithm of each dataset. Table 3 shows the data validation results consisting of accurate predictive value, and error rate.

| Table 1. Description of Datasets. | Attribute (A) | Class. (C) | Instance (I) |
|----------------------------------|---------------|-----------|-------------|
| Iris                             | 4             | 3         | 150         |
| Glass                            | 9             | 7         | 214         |
| Wine                             | 13            | 3         | 178         |

| Table 2. Attribute Weight of Datasets. |
|---------------------------------------|
| Iris | Glass | Wine |
| A1  | 0.981 | A10  | 0.336 |
| A2  | 0.745 | A2   | 0     |
| A3  | 0.676 | A3   | 0.398 |
| A4  | 0.447 | A4   | 0     |
| A5  | 0.290 | A5   | 0     |
| A6  | 0.984 | A6   | 0.978 |
| A7  | 1     | A7   | 1     |
| A8  | 1     | A8   | 1     |
| A9  | 0     | A9   | 0     |
Table 3. Accuracy of Prediction.

| Dataset | LVQ-Classifier | PSOFW-LVQ | LVQ-Classifier | PSOFW-LVQ |
|---------|----------------|-----------|----------------|-----------|
| Iris    | 96.00%         | 97.33%    | 4.00%          | 3.27%     |
| Glass   | 72.55%         | 77.06%    | 27.45%         | 22.94%    |
| Wine    | 90.98%         | 98.33%    | 9.02%          | 1.67%     |

6. Summary
In this paper, to form an optimal classification model used two methods namely, LVQ and PSO. The result of data processing and validation shows that there is an increase of classifier accuracy performance if it is weighted on attribute of each dataset. This improvement in accuracy shows that the PSO algorithm is capable of finding attribute weights that increase LVQ-classifier accuracy, although LVQ-Classifier itself has produced a fairly good classification model. Experiments conducted on this paper is devoted to continuous values, future will be tried for non-continue values.

References
[1] Mehmed K 2003 Data Mining: Concept, Models, Methods, And Algorithms IEEE Press (445 Hoes Lane, Piscataway, NJ 08854)
[2] Witten I H, Frank E and Hall M A 2011 Data Mining: Practical Machine Learning Tools and Techniques Morgan Kaufmann Publisher. Elsevier.Inc (Burlington, MA 01803, USA)
[3] Millonas M 1994 Swarms, Phase Transition, and Collective Intelligence
[4] Cardic C 1993 Proc.Int.Conf. on Machine Learning
[5] Brownlee J 2011 Clever Algorithms: Nature-Inspired programming Recipes Creative Commons Attribution-Noncommercial-Share Alike 2.5 (Australi License) ISBN: 978-1-4467-8506-5
[6] Xue B, Zhang M and Browne W N 2012 Particle Swarm Optimization for Feature Selection in Classification: A Multi Objective Approach IEEE Transaction On Cybernetics Journal. IEEE Press
[7] Kennedy J and Eberhart R 1995 Proc. IEEE Int.Conf. Neural Network vol. 4
[8] Shi Y and Eberhart R 1998 Proc. IEEE Int.CEC pp. 69-73.
[9] Kohonen T 1988 Self-Organization and Associative Memory Springer-Verlag (New York)
[10] Liu Y, Wang G, Chen H and Dong H 2011 J.Bionic Eng., vol.8 2 pp.191-200
[11] Alexandre S and Leandro N d C 2013 A Constructive Data Classification Version of the Particle Swarm Optimization Algorithm Hindawi Publishing Corporation Mathematical Problems in Engineering