# TRUST-TECH Assisted GA-SVM Ensembles and Its Applications

Yong-Feng Zhang[1(✉)], Hsiao-Dong Chiang[2], Yun-Fei Qu[3], and Xiao Zhang[1]

[1] University of Jinan, Jinan 250022, SD, China
cse_zhangyf@ujn.edu.cn
[2] Cornell University, Ithaca, NY 14853, USA
[3] State Grid Baoding Electric Power Supply Company, Baoding 071000, HB, China

**Abstract.** A framework of Genetic Algorithm-Support Vector Machine (GA-SVM) is proposed for SVM parameters (model) selection, and clustering algorithm is also integrated with the framework to generate multiple optimal models, as well as being condition of convergence for GA. Moreover, an ensemble method on various SVM models assisted by TRUST-TECH methodology is put forward, to enhance the generalization ability of a single SVM model. The performance of GA-SVM and ensemble method is testified by applying them in both classification and regression problems. Results show that, comparing with traditional parameters selection method (such as grid search), the proposed GA-SVM framework and ensemble strategy can solve general classification and regression issues more efficiently and automatically with better performance.

**Keywords:** Support vector machine · Genetic algorithm · Clustering algorithm · SVM model selection · Ensemble

## 1 Introduction

Support Vector Machine (SVM) has become a competitive learning algorithm since it was developed by Vapnik and his co-workers in 1995 [1]. For recent decades, abundant literature has proven its effectiveness in solving classification problems in pattern recognition and regression analysis, which could be partially attributed to that SVM is founded on the basis of Structural Risk Minimization (SRM) [2, 3]. Traditional machine learning algorithms, such as Artificial Neural Network (ANN), form the learning rules based on the principle of Empirical Risk Minimization (ERM), i.e., the minimization of training errors. However, SRM balances ERM and VC dimension to ensure better generalization ability. The VC dimension denotes the learning ability of a set of functions, or informally, the higher the VC dimension is, the more complicated the learning machine becomes.

One major advantage of SVM is that the optimal solution is unique when model parameters are fixed, since the constructing/construction of SVM model equals solving a convex quadratic programming problem. However, challenges exist at least in these


H. Zhang et al. (Eds.): NCAA 2022, CCIS 1637, pp. 100–114, 2022.
https://doi.org/10.1007/978-981-19-6142-7_8

fields: 1) selection of model parameters are usually empirical; 2) training process, especially for huge data, can be computationally expensive. The Sequential Minimization Optimization (SMO) [4] algorithm is one of the most popular methods in solving the QP problems in SVM and affirms its valid in many aspects and applications. Hence, we focus more on the first challenge in this paper. Selection of model parameters, (or referred as model selection, tuning parameters) aims to find the optimal parameters that minimize the generalization errors of the SVM model, and these errors can often be estimated either by direct testing based on data which have not been used for training or by given bounds in previous studies such as [5–7]. Selection of parameters has always been intractable since multiple parameters should be tuned simultaneously and searching space of parameters could be problems-depended in most cases. Usual applicable strategies include grid search [8], heuristic algorithms [9, 10], and gradient-based methods introduced in [11]. The Grid search method is always exhaustive and its accuracy may depend on the steps of the 'grid', while traditional GA or SA may be stuck in a local optimal solution and be time-consuming for convergence.

Another strategy for improving generalization ability is the ensemble of several SVM models. Literature [12] defines the ensemble of classifiers as 'a set of classifiers whose individual decisions are combined in some way (typically by weighted or unweighted voting) to classify new examples'. The literature also illustrates the reason why ensembles could often outperform individual classifiers from views of statistics, computations and representations. The ensemble of SVM actually consists of two tasks: 1) generation of different SVM models; 2) combination of these models in an appropriate manner. Literature [13] describes representative methods of both tasks and discusses their performance based on simulation results.

In this paper, we put forward a novel SVM ensemble method by adopting GA-combined clustering algorithm and TRUST-TECH technologies [14, 15]. GA combined with SVM (or GA-SVM) hybrid algorithm has shown its superior ability compared with conventional SVM applications [16, 17]. Moreover, the concept of clustering is introduced in SVM parameters tuning to ensure the diversity of SVM models as an ensemble, as well as being a criterion to end the evolution of GA. TRUST-TECH was developed to find high-quality solutions for general nonlinear optimization problems. It has been successfully applied to solve machine learning problems including optimal training ANNs, and in this paper it is adopted for optimal ensemble of SVM models.

## 2 Preliminaries

### 2.1 Support Vector Machine

SVM was first introduced to solve data classification problems or pattern recognition (Support Vector Classification, SVC), and the maximal margin classifier forms the strategy of the first SVM, namely to find the maximal margin hyperplane in an appropriately chosen kernel-induced feature space [18]. For a given training set of pairs $(x_i, y_i)$, where the instance $x_i \in R^n$ and the label $x_i \in \{1, -1\}$, a typical representation of the

implementation is given as (1)

$$\begin{aligned} \min_{\omega,b,\xi} \quad & \tfrac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}\xi_i \\ s.t. \quad & y_i(\omega^T\phi(x_i)+b) \geq 1-\xi_i, \\ & \xi_i \geq 0. \end{aligned} \tag{1}$$

Here, $C$ is the penalty parameter of the error and is always positive. $\xi_i$ is the slack variable which means allowed errors. Kernel function $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$ can map non-separable data (input space) into a higher dimension (feature space) where these mapped data would be separable, and usual kernels include radial basis function (RBF) $K(x_i, x_j) = \exp(-\gamma\|x_i - x_j\|^2)$, $\gamma > 0$ linear one $K(x_i, x_j) = \phi(x_i)^T\phi(x_j)$, and polynomial one $K(x_i, x_j) = (\gamma {x_i}^T x_j + \gamma)^d$, $\gamma > 0$, where $\gamma$, $d$ are kernel parameters. Therefore, the free parameters for classification in a given case consist of $C$ and kernel parameters.

SVM for regression analysis (or termed Support Vector Regression, SVR) is developed, for example in $\varepsilon$-SVM to find a function $f(x)$ that allows a deviation less than $\varepsilon$ from the actual target $y_i$ for all training data, and meanwhile make it as flat as possible. The expression is given by (2)

$$\begin{aligned} \min_{\omega,b,\xi} \quad & \tfrac{1}{2}\omega^T\omega + C\sum_{i=1}^{l}(\xi_i+\xi_i^*) \\ s.t. \quad & y_i - \omega^T\phi(x_i) - b \leq \varepsilon + \xi_i, \\ & \omega^T\phi(x_i) + b - y_i \leq \varepsilon + \xi_i^*, \\ & \xi_i, \xi_i^* \geq 0. \end{aligned} \tag{2}$$

Compared with (1), the additional parameter $\varepsilon$ means that a deviation less then $\varepsilon$ would be ignored or could be described as the so-called $\varepsilon$-insensitive loss function $|y_i - f(x_i)|_\varepsilon$ as (3) and Fig. 1 presented.

$$|y_i - f(x_i)|_\varepsilon = \begin{cases} 0 & |y_i - f(x_i)| - \varepsilon \\ |y_i - f(x_i)| - \varepsilon, & Otherwise \end{cases} \tag{3}$$

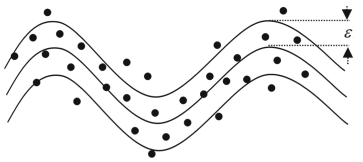


**Fig. 1.** $\varepsilon$-insensitive loss function

Lots of tools have been developed to solve SVM problems, among which LIBSVM is popular for its high efficiency in SVM classification, regression, probability estimation and other tasks [19]. LIBSVM is also adopted in the following study.

### 2.2 Genetic Algorithms

Genetic Algorithms (GA), are stochastic methods based on imitating Darwinian natural selection and genetics in biological systems, which have been successfully used for globally researching and optimizing problems [20]. In optimization problems, a set of candidate solutions is termed as population and is coded by specific chromosomes. And all candidate solutions will be evaluated by the objective function (often problems-based) to find the best solutions which can be used to build the next generation of the candidate ones. The successive steps could be described as follows: initialization of the first generation, evaluation of the candidate solutions, selection of the best ones and creation of the next generation using specific genetic operators (crossover, mutation, etc.). The iterative computation will be stopped when it comes to meeting the termination condition [9].

Although it has been applied in many fields, GA still gets its drawbacks: weak ability in local optimal search, premature convergence in certain situations, random solutions, and all these may make this algorithm time-consuming and difficult to guarantee its convergence. Thus, one aim of improvement on GA is how to speed up the convergence to get optimal solutions.

### 2.3 GA-Combined Clustering

Clustering is a common technique for statistical data analysis, and its main task is to group a set of objectives in a way that they are more similar within the same group compared with objectives in others. General clustering algorithms include K-means, K-modes and their variations.

Iterative Self-organizing Data Analysis Techniques Algorithm (ISODATA) [21] is developed based on K-means and introduces the operation of ‘splits’ and ‘merges’ on clusters, so the number of clusters is also variable. Basic processes could be summarized as: 1) Initialization of controlling parameters; 2) Randomly placing the cluster centers, and assigning samples to clusters based on their distance to cluster centers; 3) Calculation of standard deviation within each cluster and the distance between cluster centers, and splitting or merging clusters; 4) A second iteration is performed in the new cluster centers; 5) Termination conditions include the average inter-center distance which falls below the user-defined threshold, the average change in the inter-center distance between iterations that is less than a threshold, or the maximum number of iterations reached.

At the end of the evolution process of GA, a set of optimal solutions are obtained. Evaluation of these optimal solutions utilizing clustering method (ISODATA is preferred in this paper) is to analyze the probability of grouping strategy termed GA-combined Clustering Algorithm (GACA). The purpose of GACA could be addressed as: firstly, this method provides a termination condition for GA, avoiding potential time-consuming evolution process with little effect; secondly, this method describes the distinctions of optimal solutions generated by GA, and is meaningful when multi-local optimal solutions are needed for further work (e.g., ensemble of multiple solutions).

### 2.4 TRUST-TECH Technology

TRUST-TECH was developed to find high-quality solutions for general nonlinear optimization problems. It has been successfully applied to solve machine learning problems including optimal training ANNs [22, 23], estimating optimal parameters for finite mixture models [24], as well as solving the optimal power flow problem [25].

TRUST-TECH-based methods can escape from a local optimal solution and search for other solutions in a systematic and deterministic way. Another feature of TRUST-TECH is its effective cooperation with existing local and global methods. This cooperation starts with a global method for obtaining promising solutions. Then by working with robust and fast local methods, TRUST-TECH efficiently searches the neighboring subspace of the promising solutions for new local optimal solutions in a tier-by-tier manner. A high-quality optimum can be found from the multiple local optimal solutions.

### 2.5 Ensemble of SVMs

Ensemble on machine learning has been well studied in [12] and the literature also explains why ensemble can often perform better than any single classifier. Ensemble has also been widely applied in ANN, and can offer an effective way to alleviate the burden of tuning the parameters of a single ANN, moreover, the results show ensemble is effective in improved generalization capability [26, 27]. Factors may affect the ensemble results including accuracy and diversity of member networks [28–30] and the combination strategy used for ensemble [31, 32].

Ensemble of SVMs has also been studied, [13] expects that SVM ensemble can improve classification performance greatly than using a single one, especially in multi-classification cases, and propose to use the SVM ensemble based on the bagging and boosting techniques. Literature [2] uses the boosting technique to train each SVM and take another SVM for combining several SVMs.

In this paper, firstly, a set of SVM individuals would be trained by GA-combined clustering method; secondly, a selection of these SVM individuals would be optimally combined, to form a nonlinear optimization problem; finally, ensemble of SVM individuals will be finished by solving the optimization problem with TRUST-TECH technology.

## 3 Algorithm Procedures

### 3.1 SVM Parameters

Given the training data set and methods, SVM parameter $p$ would decide the unique SVM model. Thus, the selection of optimal parameters is actually the process of constructing SVM with best generalization ability. As mentioned in part A, Section II, free parameters in SVC include the penalty parameter $C$ and kernel parameters. If using RBF (Gaussian) kernel $K(x_i, x_j) = \exp(-\gamma \|x_i - x_j\|^2)$, $\gamma > 0$, the SVC parameters to be optimized here will be defined as $p_c = (C, \gamma)$, and meanwhile in SVR the similar definition is $p_r = (C, \varepsilon, \gamma)$ and $\varepsilon$ is the tolerable deviation.

The goal of parameters selection (optimization) is to identify proper $p_c$ or $p_r$ that makes accurate classification or regression results on unknown data. One common

method, as described in [8], suggests using a grid-search method with cross-validation. One typical cross-validation method is called $\upsilon$-folder cross-validation, which separates all training data with the purpose of forming $\upsilon$ subsets with equal size. And then, each subset (validating data) is valid using the SVM trained on the other $\upsilon$-1 subsets (training data). The operation repeats $\upsilon$ times so that each subset has been valid and got responding accuracy. The average of all $\upsilon$ accuracy could be seen as the accuracy to evaluate the selected parameters. According to [8], cross-validation can also prevent over-fitting problems. One extreme situation of the cross-validation method is the Leave-One-Out (LOO) method, where $\upsilon$ is equal to the number of instances of all training data. Thus, all instances will be valid by utilizing the LOO method. Literature [5] proves that LOO can provide an unbiased estimate on the probability of test errors for known data. Yet, the LOO method may be time-consuming, especially when the number of training instances is huge. Grid-search tries $p_c$ or $p_r$ in the potential solution space according to some simple rules, for example, an exponentially growing sequences of $C$ and $\gamma$ in [8]. This method is straightforward and easy for implementation. However, the solution space in SVR may become three dimensions and its efficiency is worrisome.

### 3.2 GA-SVM for Parameter Optimization

In this paper, the algorithm of optimizing parameters of SVM automatically by GA is termed GA-SVM. Input of this algorithm is normally an approximate interval of parameters, and output would be a set of optimal parameters. One significant element in the evolution process of GA-SVM is the fitness function, which is a predefined objective function that evaluates all individuals in each generation and then ranks them accordingly. Figure 2 gives the framework of this algorithm.

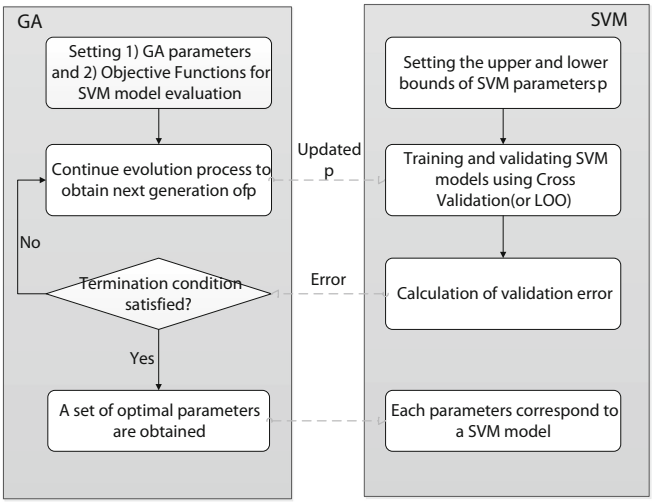


**Fig. 2.** Framework of GA-SVM

It has to be elaborated in the framework:

1) Upper and lower bounds of parameters $p$ are usually decided by a coarse cross validation process. The simplest case, searching region for $p_c$ is a closed rectangle region and $p_r$ a closed cuboid region.
2) Important GA parameters include size of the population $\alpha$ and evolution generation $\beta$. From the view of parameters selection, $\lambda$ is the number of optimal parameters in each evolution process and $\beta$ is the number of evolution times of parameters.
3) In the process of $\upsilon$-folder cross validation, let the number of all known instances is $N$, and each subset consists of $M = N/\upsilon$ instances where the correctly classified number of instances of the $j$th validating subset is $m_j$. Thus, validating error of SVC is defined as (4)

$$\sigma_c = (1 - \frac{1}{\upsilon}\sum_{j=1}^{\upsilon}\frac{m_j}{M}) * 100\% \tag{4}$$

4) Similarly, in SVR, validating error is defined as (5), which is actually the average of Mean Absolute Percentage Error (MAPE). In (5), $f_j(x)$ represents the estimation function of the $j$th validating subset, and $y_i$ is the actual value of the corresponding $x_i$.

$$\sigma_r = \frac{1}{\upsilon}\sum_{j=1}^{\upsilon}\sum_{i=1}^{M}\frac{\left|\frac{f_j(x_i)-y_i}{y_i}\right|}{M} * 100\% \tag{5}$$

5) At the beginning of the first loop, i.e., the initialization process, GA will randomly generate $\lambda$ parameters and then calculate fitness of these parameters.
6) For each individual of parameters in a certain generation, the validating error is of negative relationship with the value of fitness function (called fitness), i.e., the lower the validating errors are, the better the fitness is. By calculating the fitness of all individuals, a roulette wheel selection based nonlinear ranking method is adopted to rank all individuals according to the possibilities of the best fitness. Let the fitness function be $F_{fitness}(x)$, and $p_{\alpha a \beta b}$ be the $\alpha_a$th parameters in the $\beta_b$th generation, where $a = 1, 2, \ldots, \alpha$ and $b = 1, 2, \ldots, \beta$. $P\{x\}$is the possibility of selection for following operation of crossover and mutation. There will be (6):

$$P\{p_{\alpha_a\beta_b}\} = \frac{F_{fitness}(p_{\alpha_a\beta_b})}{\sum_{b=1}^{\beta} F_{fitness}(p_{\alpha_a\beta_b})} \tag{6}$$

7) Based on the principles of "survival of the fittest", individuals with higher fitness would reproduce more offspring. Reproduction of next generation (offspring) is realized by operations such as crossover and mutation. In this algorithm, multi-point crossover and uniform crossover are used to generate new individuals with exchanged information between old individuals. Mutation reflects another randomness in the evolution process, which generates new individuals in case that crossover is not effective. Details about the process are presented in [21].

8) In GA-SVM, the evolution process will be stopped until the $\beta$th evolution finishes. During the evolution process, parameters in each generation are recorded. At the end, all recorded parameters are compared in order to select one with the lowest validating errors as the best one to build a SVM model.
9) It has to be noted that the best parameter is not always generated at the $\beta$th evolution. Numerical studies later in this paper show the evolution process may come to convergence earlier. This explains why a fixed $\beta$ may be time-consuming with low efficiency.

### 3.3 Improvement on GA-SVM Using Clustering Method

However, drawbacks of GA-SVM for parameters selection are obvious, especially from the view of ensemble: 1) criteria for convergence is not guaranteed, because a small $\beta$ may not ensure better evolution results and a big $\beta$ may be tedious with little efforts; 2) diversity of SVM models for ensemble is not available, for that the evolution process does not consider differences of parameters within one generation and between successive generations.

Clustering method would help in GA-SVM, and in this paper we combine ISODATA with GA (or termed GACA) to optimize SVM parameters. In GACA, the evolution process begins with a much smaller generation $\widetilde{\beta}$, and after/when the evolution finishes, a set of optimal parameters are obtained. Then, ISODATA is introduced to analysis the internal relationship of parameters within this set and compared it with the previous one. If the results of clustering satisfy the termination condition, the algorithm finishes; if not, evolution process with generation $\widetilde{\beta}$ will continue based at the foundation of the last evolution results and repeat the comparing operation. The algorithm would stop until the termination condition satisfies that predefined repeating times are reached. The procedure is described in Fig. 3.

After clustering on $\{p_1\}$ finishes, GA-SVM evolution will directly generate $\{p_2\}$, i.e., comparison of clustering results happens for the first time only after clustering on $\{p_2\}$ finishes.

In the end, a set of clustered parameters $\{p_{cld}\}$ are acquired, including corresponding clusters. Moreover, in considering of Euclidean distance among parameters, the diversity of SVM models is guaranteed.

### 3.4 Ensemble on SVM Using TRUST-TECH

Let $K$ be the number of selected clusters of $\{p_{cld}\}$, and ${p_{cld}}^i$ is a representative parameter (normally the cluster center) selected from the $i$th cluster, where $i = 1, 2, \ldots, K$. Usually, ${p_{cld}}^i$ is the one with the lowest validating error of all parameters in the $i$th cluster, and all ${p_{cld}}^i$ should be within a neighborhood of $\min\{p_{cld}\}$. e.g., ${p_{cld}}^i \leq \eta \min\{p_{cld}\}$, where $\eta$ is a factor ensuring the low validating error of selected parameters. Hence, parameters with high parameters would not be chosen for the ensemble. The task of finding an optimal ensemble on SVM models is achieved by solving the following optimization problem:

$$\min \quad E(v|S, x) = \sum_{n=1}^{Q} (\sum_{m=1}^{K} v_m \circ f_m(x_n) - y_n)^2 \tag{7}$$

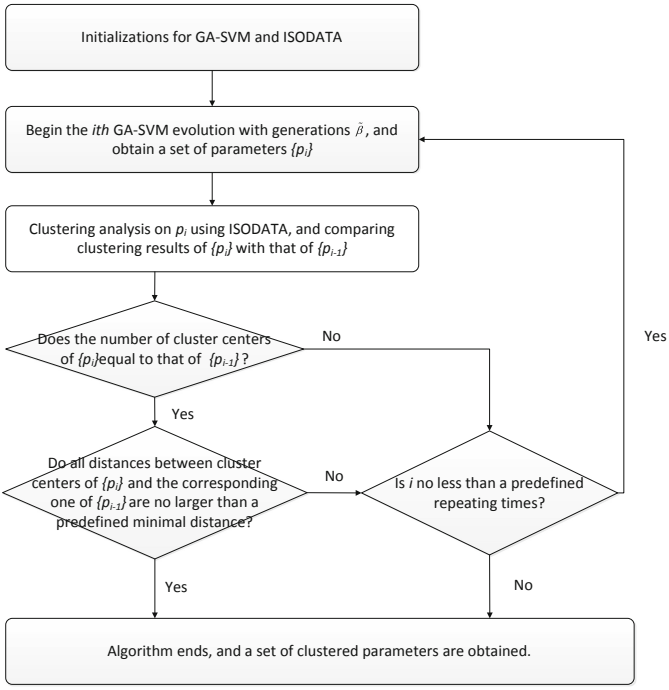


**Fig. 3.** Framework of Improving GA-SVM using ISODATA

$E(v|S, x)$ is the ensemble function under a given set of SVM models $S = \{s_1, s_2, \ldots, s_K\}$, $v$ is the rules used for combination, $\{x_n\}$ is the testing data set that is not used in the training and validating process, $Q$ is the number of samples in the testing data set, $f_m(x_n)$ is the estimation function of $m$th SVM models when input is $x_n$, and $y_n$ is the desired output. Not that in SVR output of $f_m(x_n)$ is real number, and in SVC (binary classification) only integer 1 or $-1$.

In this paper, a linear combination of SVM models is considered. Thus $v = (v_1, v_2, \ldots, v_K)$ represents the weight of each model and an optimization problem turns to a QP problem (8):

$$\begin{aligned} &\min_v \quad E(v) = \frac{1}{2} v^T C v \\ &s.t. \quad v^T e = 1, \quad v \geq 0 \end{aligned} \tag{8}$$

where $e = [1, 1, \ldots, 1]^T$, $C$ is the correlation matrix describing the relationship between different SVM models. In SVR, $C$ is calculated using (9)

$$C_{ij} = \frac{1}{Q} \sum_{k=1}^{Q} (f_i(x_k) - t_k)(f_j(x_k) - t_k), \qquad i, j = 1, 2, ..., K \tag{9}$$

While in SVC, $C$ is calculated using (10)

$$C_{ij} = \frac{1}{Q}\sum_{k=1}^{Q}\sigma_i\sigma_j, \qquad i,j = 1,2,...,K \tag{10}$$

where $\sigma_i$ denotes the validating error of the $i$th SVM tested on a testing data of $Q$ samples.

Since $C$ does not always have positive eigenvalues, the quadratic optimization problem (8) may be non-convex. Therefore, it might have multiple local optimal solutions. Traditional iterative methods (e.g., Interior Point Method, IPM) which are effective in solving the convex quadratic optimization problem may get stuck in a local optimal solution in case of multiple local optimal solutions. TRUST-TECH technology is used to solve (8), and a high-quality solution will be adopted for ensemble.

Using the logarithmic barrier function, the augmented Lagrange function of (8) is

$$L_\mu(v,\theta) = \frac{1}{2}v^T Cv \quad + \theta(v^T e - 1) + \mu\sum_{i=1}^{K}\ln v_i \tag{11}$$

where $\mu$ is the barrier parameter. Hence, the Karush-Kuhn-Tuker's optimal conditions are

$$\frac{\partial L_\mu}{\partial v} = Cv + \theta e - \mu V^{-1}e = 0 \tag{12}$$

$$\frac{\partial L_\mu}{\partial \theta} = v^T e - 1 = 0 \tag{13}$$

where $V = \mathrm{diag}(v_1, v_2, \ldots, v_K)$. Multiplying both sides of (12) with $V$, we have

$$H_\mu(v,\theta) = \begin{pmatrix} VCv + \theta Ve - \mu e \\ V^T e - 1 \end{pmatrix} = 0 \tag{14}$$

By solving (12) and (13) with decreasing $\mu \to 0$, IPM will provide a local optimal solution to the original problem (8) where $\mu = 0$.

Actually, IPM plays a role of local solver, and then TRUST-TECH is used to compute multiple local optimal solutions following procedure below:

Step1: Initialization. Set the initial point $v_0 = (1/K, 1/K, \ldots, 1/K)$ and the set of SEPs $V_s = \emptyset$.

Step2: Calculate the correlation matrix $C$ and compute its eigenvalues $\lambda_1, \lambda_2, \ldots, \lambda_K$.

Step3: Use $v_0$ as the initial point, apply the IPM to solve (8) and get an SEP $\upsilon_{s0}$, and update $V_s = \{\upsilon_{s0}\}$.

Step4: If $\min_{i=1}{}^K\lambda_i < \rho$ ($\rho$ is a very small positive value), calculate the search direction $\left\{\vec{d}_1, \vec{d}_2, ...\vec{d}_K\right\}$.

Step5: For $i = 1{:}m$, search for an exit point $\upsilon_e$ along $\vec{d}_i$ in the generalized gradient system (15),

$$\frac{dx}{dt} = -\nabla H^T(x)H(x) \tag{15}$$

which is defined based on problem (14) with associate energy function

$$E(x) = \frac{1}{2}\|H(x)\|^2 \tag{16}$$

If $\upsilon_e$ along $\vec{d}_i$ is found, step forward along the search direction to the point $\upsilon' = \upsilon_{s0} + \tau(\upsilon_e - \upsilon_{s0})$ with $\tau$ being a small positive value. $\upsilon'$ will lie in the stability region of the neighboring SEP.

Then, use $\upsilon'$ as an initial point, apply the IPM to get the tier-1 SEP, denoted as $\upsilon_{si}$, lying in the neighboring stability region, and update $V_s$ as $V_s = V_s \cup \{\upsilon_{si}\}$.

Step6: The optimal combination vector is obtained as $\upsilon^* = \text{argmin}_\upsilon\{E(\upsilon)|\upsilon \in V_s\}$.

End

The final ensemble function of SVR is expressed as (17)

$$Y(x_n) = \sum_{m=1}^{K} v_m f_m(x_n) \qquad m = 1, 2, \ldots K \tag{17}$$

where $Y(x_n)$ is the final regression results with input $x_n$.

In the case of SVC (binary classification), a weight-based voting process will determine the final results as (18)

$$Y(x_n) = \begin{cases} 1 & \sum_{m=1}^{K} v_m f_m(x_n) > 0 \\ -1 & \sum_{m=1}^{K} v_m f_m(x_n) < 0 \\ f_{m'}(x_n) & \sum_{m=1}^{K} v_m f_m(x_n) = 0 \end{cases} \tag{18}$$

where $f_{m'}(x)$ is the estimation function corresponding to the maximum $v_m \in \upsilon^*$. For multiple-class problems, a similar voting strategy is adopted, where the basic principle is that the instance is tagged to the most voted classification, and if two or more classes win the same vote, the estimation with the highest weight reaches the final decision.

## 4 Application of Algorithm in Regression

In this chapter, we will solve a regression problem by using GA-SVM and ensemble. Instead of clustering algorithms, independent repeating of GA-SVM will be applied to generate multiple SVM models, and then all these models are ranked where top ones are selected for ensemble.

### 4.1 Problem Definition

[33] points out that among all SVR problems, financial data and electric load time series prediction appear to be the worthiest topics. In this thesis, the load forecasting problem in EUNITE Network Competition 2001 is studied. According to [34], the problem is described as follows.

Given Data:

- *Electricity load demand recorded every half hour, from 1997 to 1998.*
- *Average daily temperature, from 1995 to 1998.*
- *Dates of holidays, from 1997 to 1999.*

Task:

- *prediction of maximum daily values of electrical loads for January 1999*

Main evaluation on results:

$$MAPE = 100 * \frac{\sum_{i=1}^{n} \left| \frac{L_i - \hat{L}_i}{L_i} \right|}{n}, n = 31 \tag{19}$$

To make the final results comparable, raw data are preprocessed similar to [34].

Unlike parameters in SVC, the ones in SVR are a three- dimension $p_r = (C, \varepsilon, \gamma)$ and thus a grid-search method will be more computationally complex (tries of solution increasing from $n^2$ to $n^3$, where $n$ is the number of 'grid'). Anyway, GA-SVM deals with this problem similar to SVC.

### 4.2 Multiple Models Generated by GA-SVM

In this stage, GA-SVM is firstly repeated $\rho$ times and it will get $\rho$ independent optimal parameter. According to [2], SVM models with smaller $\omega$ tend to be of higher generalization ability. Then, all $\rho$ parameters would be ranked according to $\omega$ in descending orders, and the top $\rho'$ parameter will be used for ensemble.

In the electric load forecasting problems, 50 optimal parameters (or models $S = \{s_1, s_2, \ldots, s_{50}\}$) are generated by repeating GA-SVM independently and then ranked. Top 10 models $S' = \{s_{39}, s_{49}, s_{47}, s_7, s_{50}, s_{10}, s_{18}, s_{23}, s_{21}, s_{17}\}$ are selected further according to the value of $\omega$, as showing in Table 1. By solving the corresponding optimization problem (8), weight of each model of $S'$ is obtained, denoted as $v' = (0.0365, 0, 0.4845, 0.2358, 0.2432, 0, 0, 0, 0, 0)$. Thus, the final prediction is (20).

$$Y(x_i) = \sum_{S_i \in S'} f_j(x_i) v'_j, i = 1, 2, \ldots 31, j = 1, 2, \ldots 10 \tag{20}$$

where $x_i$ represents the maximum electric load of the $i$th date in January, 1999. $f_j(x)$ is the corresponding estimation function of the $j$th which is the selected model in $S'$ with $v'_j$, its weight.

To illustrate the performance of ensemble, a comparison between the prediction results of ensemble models, single model (with the best $\omega$) and the actual electric loads is given in Fig. 4. Final results show that with the lowest $\omega$ model the *MAPE* is 2.566, while with ensemble the *MAPE* is 1.866, which is better than the result obtained by Lin in EUNITE competition using SVM with the $MAPE = 1.982$.

**Table 1.** SVM models for ensemble

| Before ranking | | After ranking | | |
|---|---|---|---|---|
| Models no | Validating error | Models no | $\omega$ (descending order) | Validating error |
| 1 | 1.833 | 39 | 523.4863 | 2.756 |
| 2 | 1.604 | 49 | 4781.672 | 1.578 |
| 3 | 1.702 | 47 | 5349.222 | 1.557 |
| 4 | 1.818 | 7 | 7717.754 | 1.683 |
| 5 | 1.863 | 50 | 9073.668 | 1.590 |
| 6 | 1.510 | 10 | 9114.403 | 1.665 |
| 7 | 1.683 | 18 | 16202.6 | 1.563 |
| 8 | 1.361 | 23 | 16685.5 | 1.599 |
| 9 | 1.699 | 21 | 20775.92 | 1.568 |
| 10 | 1.665 | 17 | 26607.92 | 1.714 |
| 11 | 1.762 | 34 | 30603.84 | 1.617 |
| 12 | 1.828 | 2 | 30905.61 | 1.604 |
| 13 | 1.825 | 43 | 31191.94 | 1.586 |
| 14 | 1.407 | 27 | 55406.93 | 1.724 |
| 15 | 1.839 | 16 | 63099.55 | 1.610 |
| … | … | … | … | … |
| 50 | 1.610 | 42 | 68794.02 | 1.754 |

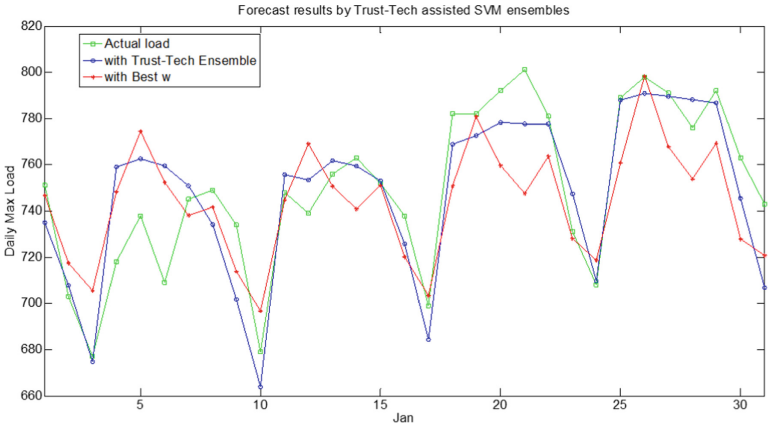


**Fig. 4.** Comparison among ensemble, single and actual load

## 5 Conclusion and Future Work

A framework of GA-SVM is proposed in this paper, which will automatically optimize parameters of SVM and generate optimal SVM models. The clustering algorithm is also combined with GA-SVM as an enhancement, providing devious SVM models for future ensemble, as well as being a convergent criterion. This paper also proposes an ensemble strategy with the help of TRUST-TECH. Moreover, experiments in both classification and regression problems show the validity of GA-SVM, the clustering algorithm and the ensemble.

## References

1. Cortes, C., Vapnik, V.: Support-vector networks. Mach. Learn. **20**(3), 273–297 (1995)
2. Vapnik, V.: The Nature of Statistical Learning Theory. Springer, New York (2000). https://doi.org/10.1007/978-1-4757-3264-1
3. Vapnik, V.: An overview of statistical learning theory. IEEE Trans. Neural Netw. **10**(5), 988–999 (1999)
4. Platt, J.: Sequential minimal optimization: a fast algorithm for training support vector machines (1998)
5. Vapnik, V., Chapelle, O.: Bounds on error expectation for support vector machines. Neural Comput. **12**(9), 2013–2036 (2000)
6. Chang, M.W., Lin, C.J.: Leave-one-out bounds for support vector regression model selection. Neural Comput. **17**(5), 1188–1222 (2005)
7. Chapelle, O., Vapnik, V.: Model selection for support vector machines. In: Advances in Neural Information Processing Systems, pp. 230–236 (1999)
8. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification (2003)
9. Huang, C.L., Wang, C.J.: A GA-based feature selection and parameters optimization for support vector machines. Expert Syst. Appl. **31**(2), 231–240 (2006)
10. Pai, P.F., Hong, W.C.: Support vector machines with simulated annealing algorithms in electricity load forecasting. Energy Convers. Manag. **46**(17), 2669–2688 (2005)
11. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Mach. Learn. **46**(1), 131–159 (2002)
12. Dietterich, T.G.: Ensemble methods in machine learning. In: Kittler, J., Roli, F. (eds.) MCS 2000. LNCS, vol. 1857, pp. 1–15. Springer, Heidelberg (2000). https://doi.org/10.1007/3-540-45014-9_1
13. Kim, H.C., Pang, S., Je, H.M., Kim, D., Bang, S.Y.: Constructing support vector machine ensemble. Pattern Recogn. **36**(12), 2757–2767 (2003)
14. Zhang, Y.F., Chiang, H.D.: A novel consensus-based particle swarm optimization-assisted trust-tech methodology for large-scale global optimization. IEEE Trans. Cybern. **47**(9), 2717–2729 (2017)
15. Zhang, Y.F., Chiang, H.D., Wang, T.: A novel TRUST-TECH-enabled trajectory-unified methodology for computing multiple optimal solutions of constrained nonlinear optimization: theory and computation. IEEE Trans. Syst. Man Cybern. Syst. **52**(1), 473–484 (2022)
16. Nguyen, T., Gordon-Brown, L., Wheeler, P., Peterson, J.: GA-SVM based framework for time series forecasting. In: 5th International Conference on Natural Computation, pp. 493–498. IEEE (2009)
17. Wang, L., Xu, G., Wang, J., Yang, S., Guo, L., Yan, W.: GA-SVM based feature selection and parameters optimization for BCI research. In: 7th International Conference on Natural Computation, pp. 580–583. IEEE (2011)

18. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines and Other Kernel-Based Learning Methods. Cambridge University Press, Cambridge (2000)
19. Chang, C.C., Lin, C.J.: LIBSVM: a library for support vector machines. ACM Trans. Intell. Syst. Technol. (TIST) **2**(3), 1–27 (2011)
20. Srinivas, M., Patnaik, L.M.: Genetic algorithms: a survey. Computer **27**(6), 17–26 (1994)
21. Ball, G. H., Hall, D. J.: ISODATA, a novel method of data analysis and pattern classification. Stanford Research Inst., Menlo Park, CA (1965)
22. Wang, B., Chiang, H.D.: ELITE: ensemble of optimal input-pruned neural networks using TRUST-TECH. IEEE Trans. Neural Netw. **22**(1), 96–109 (2007)
23. Zhang, Y.F., Chiang, H.D.: Enhanced ELITE-load: a novel CMPSOATT methodology constructing short-term load forecasting model for industrial applications. IEEE Trans. Ind. Inf. **16**(4), 2325–2334 (2020)
24. Reddy, C.K., Chiang, H.D., Rajaratnam, B.: Trust-tech-based expectation maximization for learning finite mixture models. IEEE Trans. Pattern Anal. Mach. Intell. **30**(7), 1146–1157 (2008)
25. Chiang, H.D., Wang, B., Jiang, Q.Y.: Applications of TRUST-TECH methodology in optimal power flow of power systems. In: Kallrath, J., Pardalos, P.M., Rebennack, S., Scheidt, M. (eds.) Optimization in the Energy Industry, pp. 297–318. Springer, Heidelberg (2009). https://doi.org/10.1007/978-3-540-88965-6_13
26. Rao, N.S.V.: On fusers that perform better than best sensor. IEEE Trans. Pattern Anal. Mach. Intell. **23**(8), 904–909 (2001)
27. Zhou, Z.H., Wu, J., Tang, W.: Ensembling neural networks: many could be better than all. Artif. Intell. **137**(1–2), 239–263 (2002)
28. Opitz, D.W., Shavlik, J.W.: Generating accurate and diverse members of a neural-network ensemble. In: Advances in Neural Information Processing Systems, pp. 535–541 (1995)
29. Brown, G.: Diversity in neural network ensembles. University of Birmingham (2004)
30. Windeatt, T.: Accuracy/diversity and ensemble MLP classifier design. IEEE Trans. Neural Netw. **17**(5), 1194–1211 (2006)
31. Hashem, S.: Optimal linear combinations of neural networks. Neural Netw. **10**(4), 599–614 (1997)
32. Ueda, N.: Optimal linear combination of neural networks for improving classification performance. IEEE Trans. Pattern Anal. Mach. Intell. **22**(2), 207–215 (2000)
33. Sapankevych, N.I., Sankar, R.: Time series prediction using support vector machines: a survey. IEEE Comput. Intell. Mag. **4**(2), 24–38 (2009)
34. Chen, B.J., Chang, M.W.: Load forecasting using support vector machines: a study on EUNITE competition 2001. IEEE Trans. Power Syst. **19**(4), 1821–1830 (2004)