Application of MOGA Search Strategy to SVM Training Data Selection

Tomoyuki Hiroyasu[1], Masashi Nishioka[2], Mitsunori Miki[3], Hisatake Yokouchi[1]

[1] Faculty of Life and Medical Sciences, Doshisha University,
1-3 Tatara Miyakodani Kyotanabe, Kyoto, Japan
tomo@is.doshisha.ac.jp, hyokouch@mail.doshisha.ac.jp
[2] Graduate School of Engineering, Doshisha University
mnishioka@mikilab.doshisha.ac.jp
[3] Faculty of Science and Engineering, Doshisha University
mmiki@mail.doshisha.ac.jp

Abstract. When training Support Vector Machine (SVM), selection of a training data set becomes an important issue, since the problem of overfitting exists with a large number of training data. A user must decide how much training data to use in the training, and then select the data to be used from a given data set. We considered to handle this SVM training data selection as a multi-objective optimization problem and applied our proposed MOGA search strategy to it. It is essential for a broad set of Pareto solutions to be obtained for the purpose of understanding the characteristics of the problem, and we considered the proposed search strategy to be suitable. The results of the experiment indicated that selection of the training data set by MOGA is effective for SVM training.

1 Introduction

Support Vector Machine (SVM) is a pattern classification technique introduced by V. Vapnik et al. [1]. The basic idea of SVM is to map an input vector x into a high dimensional feature space H by Φ and construct an optimal separating hyperplane in this space [2]. SVM has been applied to various pattern recognition cases, such as digit recognition [1], text categorization [3], and face detection [4].

The goal of SVM is to achieve the best generalization performance by learning on a given training data set. For this purpose, it is important for the problem of overfitting [5] to be considered in the training of SVM. Generally, all examples in the training data set are treated equally and used in the training process of SVM, however there are examples with more information or those that can be misleading. Therefore, there are existing researches on data categorization to group examples based on their usefulness [6].

With this background, we considered to handle the selection of training data set for SVM as a multi-objective optimization problem, and solve it by applying multi-objective genetic algorithms (MOGAs). There are researches on application of multi-objective optimization to SVM in forms of evolutionary SVM [5]. As these researches focus on the learning mechanism of SVM, our approach is to optimize the training data set using MOGAs.

There are two objectives to this training data selection problem, which are training error and the confidence margin [6]. Training error is to be minimized, and confidence margin is to be maximized in this case. Minimization of the training error is likely to result in overfitting, whereas maximizing the confidence margin prevents overfitting, and a trade-off relationship is expected between these two objectives. One characteristic of this training data selection problem is the importance of the optimal solution of each objective. In order to provide a decision maker with good understanding of the trade-off relationship between two objectives, extreme solutions must be included in the final Pareto solutions.

There are many multi-objective genetic algorithms (MOGAs) developed to date [7, 8, 9, 10, 11] with the purpose to find Pareto optimal solutions. In multi-objective optimization, it is desirable for the obtained solutions to be high quality regarding accuracy, uniform distribution, and broadness. Accuracy is how close the obtained solutions are to the true Pareto front, and uniform distribution is how evenly located the solutions are without concentrating in certain areas. Broadness is how widespread the solutions are and is decided by the optimal solutions of each objective located at the edge of the Pareto front.

Many MOGAs have mechanisms to improve accuracy and uniform distribution of the solutions. However, not many mechanisms are available to improve broadness of the solutions. NSGA-II [10] and SPEA2 [11] are two well-known MOGAs today, but both algorithms only have mechanisms included to preserve the obtained broadness of the solutions. Same can be said about other algorithms as well, and few are capable of improving broadness of the solutions.

As formerly mentioned, it is important for broad solutions to be obtained when understanding characteristics of the optimization problem. For this reason, Okuda et al. proposed the Distributed Cooperation Scheme [12], which utilizes single-objective GA (SOGA) along with MOGA. SOGA is utilized to search for the optimal solutions of each objective, which leads to improvement of broadness. It was confirmed that the Distributed Cooperation Scheme is capable of deriving broader solutions than conventional MOGAs. However, preliminary experiments have also indicated that the convergence speed is reduced because the solutions are broadened from the beginning of the search.

Because it is difficult to simultaneously improve convergence and broadness of the solutions, we consider dividing the search into two phases in our proposed search strategy. The first phase in the proposed search strategy is to improve convergence of the solutions, and the second is to improve the broadness of the solutions. This search strategy is capable of deriving broader solutions compared to conventional MOGAs without deterioration of accuracy. Therefore, we consider applying this search strategy to the selection of SVM training data.

In this paper, we first introduce our proposed search strategy consisting of two search phases. The search strategy is tested on test problems to verify its performance compared to conventional MOGAs. Then we adapted this search strategy to SVM training data selection problem.

2 Search Strategy for Multi-objective Genetic Algorithm with Consideration of Accuracy and Broadness

2.1 Importance of Broadness

The search strategy we propose considers accuracy and broadness of the solutions. Although, conventional MOGAs attempt to derive Pareto optimal solutions, there are not many mechanisms to improve broadness of the solutions. Lack of broadness becomes a problem especially in real-world optimization problems where a decision maker selects a solution based upon the given solutions. It is important to understand the possible range of solutions for the problem, and deriving solutions in a limited portion of the Pareto front is not enough. Obtained solutions cannot be considered to be as broad as possible without a mechanism to actively improve broadness. Therefore, it is essential for a broadness improving mechanism to be included in the search strategy.

The proposed search strategy consists of two search phases as shown in Fig. 1. The first phase is a search to improve convergence, and the second is for broadness. The search phases are in this order, as the final solutions obtained must be comparable to conventional MOGAs regarding accuracy and also be broad. Especially, in cases where the search time is limited, it becomes important to ensure the accuracy of the solutions first.

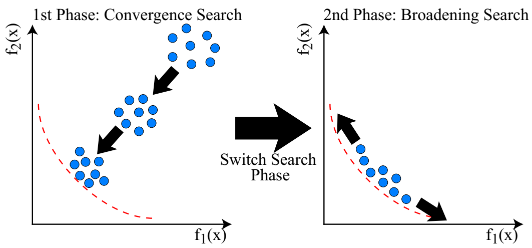


Fig. 1. Concept of the Proposed Search Strategy

2.2 1st Phase: Convergence Search

In the convergence search, preference of a decision maker is adopted in form of a reference point [13]. This reference point can be located in both feasible and infeasible regions. Conventional MOGAs base their search on the dominance relationship of the solutions, but the proposed search method bases its search on the distance information. That is, solutions closer to the reference point are prioritized in the search, which leads to convergence of the solutions around the reference point. The concept of this search is illustrated in Fig. 2.

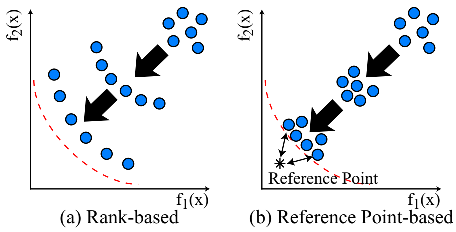


Fig. 2. Concept of the Reference Point-based Search

The proposed search strategy is based on conventional MOGAs, and the distance information is utilized in the selection criterion of the mating selection. The mating selection method is described below, and the archive size here is N.

Step 1: Sort archive solutions in ascending order of the Euclidean distance from the reference point.
Step 2: Add top $\frac{N}{2}$ solutions to the search population.
Step 3: Select remaining solutions by tournament selection based on their rank. If multiple solutions with same rank exist, select the solution with the smallest Euclidean distance.

$\frac{N}{2}$ closest solutions to the reference point are copied to the search population in Step 2, because these solutions are not guaranteed to be selected using methods such as tournament selection. Copying these solutions to the search population should result in improvement of convergence. In addition, both rank and Euclidean distance are considered in the tournament selection at Step 3, which allows selection of non-dominated solutions close to the reference point, and the search is directed towards the reference point while preserving diversity.

2.3 2nd Phase: Broadening Search

The Distributed Cooperation Scheme of Okuda et al. [12] is adopted in the broadening search. The search population is divided into subpopulations that search using MOGA and SOGA in this scheme. Henceforth, subpopulations that search with MOGA and SOGA are called MOGA population and SOGA population, respectively. When there are k objectives, the search population is divided into $k + 1$ subpopulations: one MOGA population and k SOGA populations. The concept of this scheme is illustrated in Fig. 3. As this is a scheme, any MOGA or SOGA methodology can be adopted. In this study, NSGA-II [10] and SPEA2 [11] are each adopted as the MOGA population, and DGA [14] as the SOGA population.

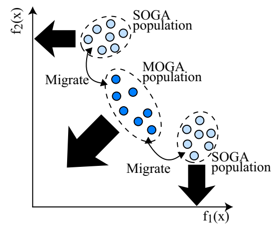


Fig. 3. Concept of the Distributed Cooperation Scheme

MOGA and SOGA populations search in a parallel manner in the Distributed Cooperation Scheme, and the best solutions from each population are exchanged every interval generations, which was set to 25 generations in this study. The best solution of the f_i SOGA population is the solution with the best f_i objective value. On the other hand, best solutions of the MOGA population are non-dominated solutions with best f_i objective value, and k solutions exist in a k-objective problem. Migration of solutions in a two-objective problem is shown in Fig. 4.

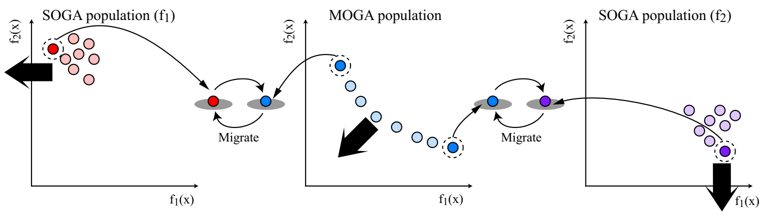


Fig. 4. Concept of Migration in the Distributed Cooperation Scheme

The algorithm of the Distributed Cooperation Scheme with population size of N in the k-objective problem is shown below.

Step 1: Randomly generate N individuals.

Step 2: Divide the individuals into MOGA and k SOGA populations with $\frac{2N}{k+2}$ individuals in MOGA population and $\frac{N}{k+2}$ individuals in SOGA population.

Step 3: Search for non-dominated solutions in the MOGA population and optimal solutions of each objective in the SOGA populations.

Step 4: Collect solutions from all populations and update archive.

Step 5: Exchange best solutions between MOGA and SOGA populations every interval generations.

Step 6: End if terminate criterion is met; else go back to Step 3.

2.4 Search Strategy

In the proposed search strategy, the convergence search described in section 2.2 is conducted, followed by the broadening search described in section 2.3. When to switch the search phase becomes important in this case. It is preferable that the search be switched when the solutions have converged. We consider the following two cases of convergence:

- Advancement of the search towards the Pareto optimal front is little.
- Newly derived non-dominated solutions contribute little to accuracy.

Therefore, we adopt two convergence indicators in switching the search phase. The first indicator is the one utilized in MRMOGA [15]. It is an average ratio of non-dominated solutions in the archive that are dominated by the derived solutions over several generations. This ratio will be high when the search is advancing and low when converged. In detail, when non-dominated solution set of the archive at the ith generation is $PF_{known}(i)$, the ratio of $PF_{known}(i)$ that is dominated ($dominated_i$) can be calculated. Based on the average ratio over g generations, it can be determined that the search has converged if criterion (1) is met.

$$\sum_{i=1}^{g} \frac{dominated_i}{g} \leq \epsilon \tag{1}$$

With MRMOGA, the value of $\epsilon = 0.05$ is used and we used this in our research for two-objective problems as well. $\epsilon = 0.025$ is used for three-objective problem, because it becomes more difficult to dominate other solutions with increasing number of objectives. Moreover, the period of g generations is set to be the same as the migration interval in section 2.3, which was 25 generations.

The second convergence indicator is the average number of archived non-dominated solutions that are dominated by each newly generated non-dominated solution. This indicator will cover the problem of MRMOGA's indicator that the number of newly derived non-dominated solutions is not considered. For example, average value of 1 means that each new non-dominated solution dominates 1 archived non-dominated solution. This indicator shows the effectiveness of the newly generated solutions for advancing the search. Lower average value means that the search is shifting to improvement of diversity instead of accuracy. Therefore, the search can be determined as converged if this indicator value is low.

We take average value of this indicator over g generations and determine that the search has converged if criterion (2) is met. Here, μ_i is the average number of archived non-dominated solutions that are dominated by each new non-dominated solution at ith generation, and we used $g = 25$.

$$\sum_{i=1}^{g} \frac{\mu_i}{g} \leq \alpha \tag{2}$$

We used $\alpha = 0.5$ for two-objective and $\alpha = 0.25$ for three-objective problems as the criterion in our research, since it showed good results in the preliminary experiments. Using the two indicators mentioned above, we switch the search phase when either criterion is met. The process of the search strategy for a k-objective problem is shown below.

Step 1: Initialize the archive.
Step 2: Conduct convergence search as described in section 2.2.
Step 3: Check criterion (1) and (2) every g generations. Go to Step 4 if either criterion is met, else go back to Step 2.
Step 4: Divide solutions stored in archive into $k+1$ populations.
Step 5: Conduct broadening search as described in section 2.3.
Step 6: End if terminate criterion is met; else go to Step 5.

3 Verification of Search Strategy's Performance

A numerical experiment was performed to verify the effectiveness of the proposed search strategy by comparison with NSGA-II and SPEA2. The MOGA methodology of the proposed search strategy is NSGA-II and SPEA2, and DGA was adopted as the SOGA population. The test problems used in this experiment were KUR [16] and multi-objective knapsack problems. KUR is a two-objective continuous problem with 100 design variables [16]. KP500-2 (i.e., 2 objectives, 500 items), KP750-2, and KP750-3 [9] were selected as multi-objective knapsack problems.

We adopted inverted generational distance (IGD) [17], hypervolume (HV) [18], and spread [19] as the metrics in this experiment: IGD is the average distance from each solution of the Pareto optimal front to the closest obtained solution, and is a metric of accuracy and broadness; HV is a metric of overall performance; and spread, calculated as the sum of differences between maximum and minimum values of each objective within the obtained Pareto front, is a metric of broadness. The Pareto optimal front must be known to calculate IGD, but is unknown for KUR, KP750-2, and KP750-3 problems. Therefore, we obtained near Pareto optimal solutions beforehand using a much greater population size and generations, and used them for IGD calculation.

For both the proposed search strategy and conventional MOGAs, population size is set to 120 for problems other than KP750-3, and 150 for KP750-3. The maximum generation is 1000, and the number of evaluations is the same for all methods. In addition, 2-point crossover is utilized with crossover rate of 1.0, and the mutation rate is 1/Chromosome Length. The parameters specific to the DGA used in the proposed search strategy are as follows: sub population size is 10, tournament selection with tournament size of 4, migration rate is 0.5, and migration interval is 5. The topology of migration is random ring. In the proposed search strategy, a reference point must be set for each problem. Several locations of reference points are tested for each problem.

3.1 Results

50% attainment surfaces of KUR, KP500-2,and KP750-2 by the proposed search strategy, NSGA-II, and SPEA2 in 30 trials are shown in Figs. 5 to 7. In these figures, the reference points of the search strategy are set as (-1000, -400), (21000, 21000), and (30000, 30000) for KUR, KP500-2, and KP750-2, respectively.

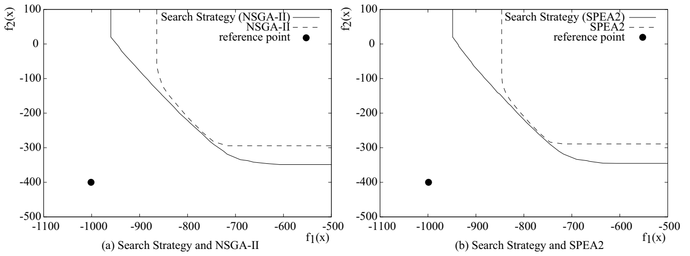


Fig. 5. 50% attainment surfaces of KUR (30 Trials)

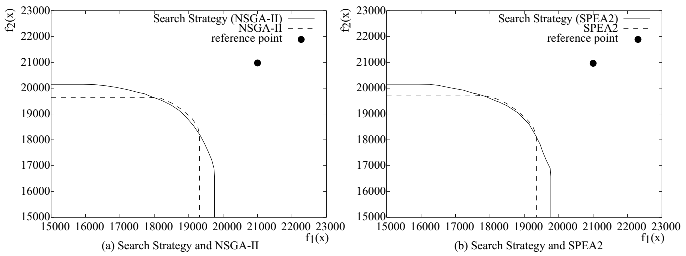


Fig. 6. 50% attainment surfaces of KP500-2 (30 Trials)

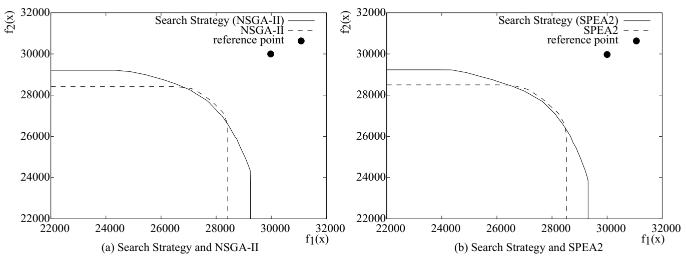


Fig. 7. 50% attainment surfaces of KP750-2 (30 Trials)

The search results in Figs. 5 to 7 indicate that the search strategy obtained broader solutions than NSGA-II or SPEA2. Broader solutions provide more information of the Pareto front, which is important especially in problems such as KUR and KP750-2 where the optimal front is unknown.

In addition, transitions of the 50% attainment surfaces of KP750-2 by the search strategy with three different reference points are shown in Fig. 8. NSGA-II is utilized in the search strategy, and the search was switched from the first phase to the second phase at the average of 600th or 575th generation depending on the reference point used.

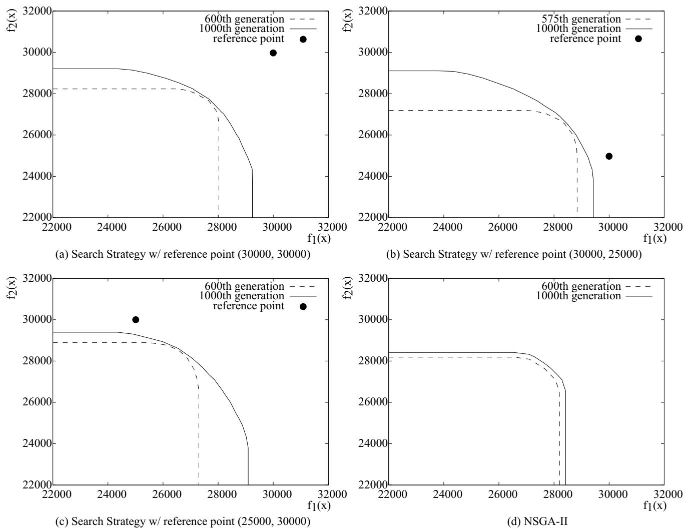


Fig. 8. Transition of the 50% attainment surfaces of KP750-2 (30 Trials)

As shown in Fig. 8, solutions converge to different regions depending on the location of the reference point. The resulting 50% attainment surfaces are biased toward the edge of the Pareto front in Fig. 8 (b) and (c), but still broader solutions are obtained compared to NSGA-II shown in Fig. 8 (d). Moreover, it can be seen that the broadness of the solutions improve greatly after the search phase is switched. Similar results were seen in other test problems as well, and these results indicate that the proposed search strategy is successful in first converging and then broadening solutions.

Next, the mean values and the standard deviation of IGD, spread, and HV are shown in Tables 1 to 3. For IGD in Table 1, the obtained solutions are closer to the Pareto optimal front when the value is close to 0. On the other hand, solutions with greater values of spread and HV are better. For the search strategy, reference points are set at (-1000, -750), (21000, 21000), (30000, 30000), and (30000, 30000, 30000) for KUR, KP500-2, KP750-2, and KP750-3, respectively.

The mean values of IGD in Table 1 indicate that both implementation of the search strategy is performing equivalent to or better than NSGA-II or SPEA2. Therefore, the search strategy is comparable to both NSGA-II and SPEA2 with regard to accuracy. IGD also describes how close the obtained solutions are to the optimal front regarding broadness. Consequently, the solutions obtained by NSGA-II and SPEA2 are not sufficiently broad.

The spread values shown in Table 2 also indicate that the search strategy obtained broader solutions. From this, it can be said that the approach to broaden solutions after converging them is successful in obtaining broad solutions. Mean HV values shown in Table 3 also show better results for the search strategy. These results indicate that the proposed search strategy is effective for maintaining accuracy comparable to conventional MOGAs and deriving broader solutions.

Table 1. Inverted Generational Distance

		KUR	KP500-2	KP750-2	KP750-3
Search Strategy (NSGA-II):	mean	0.04088	0.01311	0.01548	0.06773
	SD	0.00868	0.00112	0.00130	0.00270
NSGA-II:	mean	0.08846	0.02862	0.02853	0.07602
	SD	0.01061	0.00220	0.00167	0.00314
Search Strategy (SPEA2):	mean	0.04271	0.01322	0.01726	0.06567
	SD	0.00731	0.00099	0.00189	0.00233
SPEA2:	mean	0.10841	0.02478	0.02574	0.06655
	SD	0.01358	0.00176	0.00158	0.00210

Table 2. Spread

		KUR	KP500-2	KP750-2	KP750-3
Search Strategy (NSGA-II):	mean	682.04	6401.10	9134.77	14579.27
	SD	15.30	356.46	489.59	1520.48
NSGA-II:	mean	321.02	2497.23	3130.30	7845.73
	SD	23.06	226.85	236.35	448.87
Search Strategy (SPEA2):	mean	677.28	6568.13	9650.20	13789.67
	SD	15.90	338.40	577.90	2263.41
SPEA2:	mean	263.42	3000.57	3771.07	5176.10
	SD	25.39	200.25	316.30	412.81

Table 3. Hypervolume

		KUR	KP500-2	KP750-2	KP750-3
Search Strategy (NSGA-II):	mean	2.86E+05	3.95E+08	8.47E+08	2.41E+13
	SD	7.81E+03	1.23E+06	3.22E+06	2.89E+11
NSGA-II:	mean	2.41E+05	3.79E+08	8.06E+08	2.19E+13
	SD	6.53E+03	1.36E+06	2.79E+06	1.60E+11
Search Strategy (SPEA2):	mean	2.84E+05	3.95E+08	8.49E+08	2.40E+13
	SD	7.42E+03	1.54E+06	3.84E+06	5.45E+11
SPEA2:	mean	2.35E+05	3.81E+08	8.10E+08	2.16E+13
	SD	7.57E+03	1.47E+06	2.99E+06	1.22E+11

4 Application to SVM Training Data Set Selection Problem

When training SVM, it is recommended to select the examples to include in the training data set, because some examples are more useful than others. Data categorization is used for this purpose [6]. In this section, the proposed search strategy is applied to the optimization of training data set selection for SVM. There are two objectives to be optimized in this problem, and they are:

- Minimize training error (f_1)
- Maximize minimum confidence margin (f_2)

Training error is measured with the trained SVM on the entire data set, and is to be minimized. The confidence margin for an example $(\mathbf{x}_i, y_i)$ is measured by $y_i\tilde{g}(\mathbf{x}_i)$, where $\tilde{g}(\mathbf{x})$ is the SVM decision function [6]. We calculate the confidence margin for all examples of the data set, and considered to maximize its minimum value. Examples with negative confidence margins are excluded here, because they are mislabeled. It is expected for a trade-off to exist between these two objectives, because improvement of training error can result in overfitting, which leads to smaller confidence margin.

We applied this SVM training data selection on three data sets from the UCI machine learning repository [20]. The data sets used in this experiment are shown in Table 4. All features of the data sets are scaled to the range of [-1, 1]. For the purpose of examining the generalization performance of the trained SVM, we randomly sampled 20% of the given data set as the hold-out test set. This test set is not used in the training of SVM, therefore the number of examples available for training is 80% of the entire data set. C-SVM with RBF kernel is used, and the parameters C and γ are decided in advance using cross validation by parallel grid search [21].

Table 4. The data sets used in the experiment. n is the number of data and m is the number of features.

Data set	n	m	classes	C	γ
diabetes	768	8	2	32.0	0.03125
heart	270	13	2	2048.0	0.00049
liver-disorders	345	6	2	512.0	0.03125

This problem is designed in a similar manner to multi-objective knapsack problems, and each example from a data set is represented by 0/1 bit. The example is included in the training data set if the bit value is 1, and not if the bit value is 0. Therefore the length of a chromosome in MOGA is the same as the number of examples in the data set. By this implementation, the number of examples in the training data set and the examples included are decided at the same time.

Population size of 120 and maximum generation of 250 are used in this experiment, and the other basic parameters of the proposed search strategy are the same as the previous experiment. Individuals are initialized randomly, and the number of examples in a training data set range from 0 to the entire data set. Moreover, the reference point is set at $(0, 1)$ in this experiment, because an example with a confidence margin of less than 1 is considered to be a support vector [6].

4.1 Results

10 trials were conducted on each data set, and Figs. 9 to 11 show the 50% attainment surface of diabetes, heart, and liver-disorders data sets, respectively. Trade-off relationship is confirmed between the two objectives in all cases.

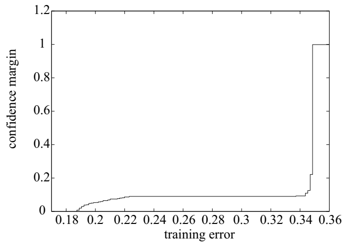


Fig. 9. 50% attainment surface of diabetes data set (10 Trials)

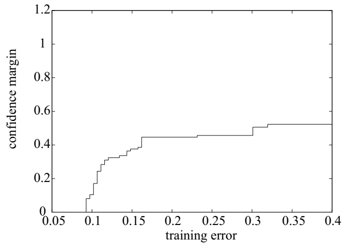


Fig. 10. 50% attainment surface of heart data set (10 Trials)

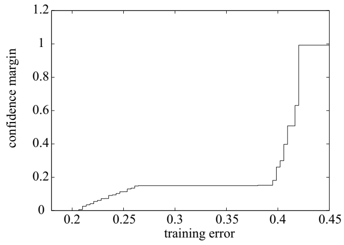
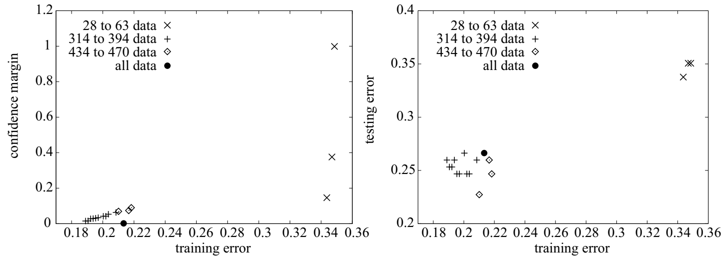


Fig. 11. 50% attainment surface of liver-disorders data set (10 Trials)

From the attainment surfaces shown in Figs. 9 to 11, it was confirmed that the improvement of training error results in smaller confidence margin, and vice versa. The Pareto fronts obtained for the diabetes and liver-disorders data sets were sparse in this experiment, which resulted in the nonsmooth front.

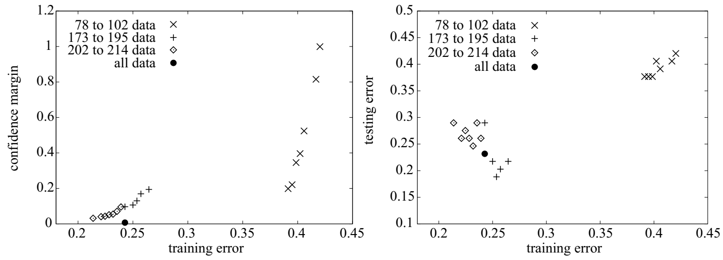
Next, we examined the generalization performance of the trained SVMs using the hold-out test set. Figs. 12 and 13 show the solutions for the diabetes and liver-disorders data set obtained in a single run. In both figures, the left figure shows the obtained Pareto solutions, and the right figure shows the generalization performance of the solutions from the left figure. The x-axis in both left and right plots show the training error, so the solutions in both plots correspond to each other on the x-axis. In addition, the solutions shown here are grouped according to the number of examples included in their training data set. The result of the SVM trained with the entire data set is plotted for comparison as well.

Figs. 12(a) and 13(a) show that selecting the training data set, rather than using the entire data set can obtain SVMs with better training error and the confidence margin. The same can be said about the generalization performance shown in Figs. 12(b) and 13(b) as well, because there are SVM models with



(a) Pareto solutions (b) Generalization performance

Fig. 12. Example of solutions obtained for diabetes data set



(a) Pareto solutions (b) Generalization performance

Fig. 13. Example of solutions obtained for liver-disorders data set

better test errors. The confidence margin value of the SVM model trained with the entire data set is small compared to other SVM models, and it is likely to be overfit. Therefore, we can understand that the selection of the training data set is beneficial.

If we compare the distribution of the solutions for diabetes data set in Fig. 12, we can see that the best SVM model regarding the training error does not perform best with the test data, which may be caused by overfitting. On the other hand, SVM models with large confidence margins showed very poor performance regarding both the training error and test error. Similar results were observed with the results of liver-disorder data set shown in Figs 13 as well. It is important for the extreme solutions to be obtained in such a case, because they provide the information on the possible range for the SVM's performance. For this reason, we consider the multi-objective approach combined with the hold-out test data to be effective for reducing the possibility of overfitting when selecting the training data set.

Another point we focused on is the number of training data used in each SVM model. Comparing the distribution of the solutions grouped according to the number of training data in Figs. 12 and 13, we observed that the training error is generally low when many examples are used in the SVM training, and high when less examples are used. Although these results show that the increasing number of training data leads to improvement of training error in general, further research is still needed. We assume that cases exist where the SVM parameters of C and γ used in this experiment are not proper for that particular training data set. Therefore, we will consider including C and γ as design variables of MOGA and optimize them for each training data set in the future research.

5 Conclusions

In this paper, we handled the selection of training data set for SVM as a multi-objective optimization problem, and applied MOGA search strategy to it. The proposed search strategy consists of two search phases to separately improve convergence and broadness of the solutions. The first phase improves the convergence of the solutions, and a reference point specified by a decision maker is adopted for this purpose. In the second phase, the solutions are broadened using the Distributed Cooperation Scheme. Through a numerical experiment, we confirmed that the proposed search strategy is capable of deriving broader solutions compared to conventional MOGAs without deterioration of accuracy.

The search strategy was applied to the SVM training data selection problem, and the results showed that there exists trade-off relationship between training error and the confidence margin. SVM models trained with selected training data showed better performance compared to the model trained with the entire data set. From this result, we confirmed the importance of selecting the training data set when training SVM. In future research, we will consider to optimize the SVM parameters along with the selection of the training data set.

References

[1] Corinna Cortes and Vladimir Vapnik. Support-Vector Networks. *Machine Learning*, Vol. 20, No. 3, pp. 273–297, 1995.

[2] Olivier Chapelle, Vladimir Vapnik, Olivier Bousquet, and Sayan Mukherjee. Choosing Multiple Parameters for Support Vector Machines. *Machine Learning*, Vol. 46, No. 1-3, pp. 131–159, 2002.

[3] Thorsten Joachims. Text Categorization with Support Vector Machines: Learning with Many Relevant Features. In *Proceedings of the European Conference on Machine Learning*, pp. 137–142, 1997.

[4] Edgar Osuna, Robert Freund, and Federico Girosi. Training Support Vector Machines: An Application to Face Detection. In *Proceedings of 1997 IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, pp. 130–136, 1997.

[5] Ingo Mierswa. Controlling Overfitting with Multi-Objective Support Vector Machines. In *GECCO '07: Proceedings of the 9th annual conference on Genetic and Evolutionary Computation*, pp. 1830–1837, 2007.

[6] Ling Li, Amrit Pratap, Hsuan tien Lin, and Yaser S. Abu-mostafa. Improving Generalization by Data Categorization. In *Knowledge Discovery in Databases: PKDD 2005, volume 3721 of Lecture Notes in Artificial Intelligence*, pp. 157–168, 2005.

[7] D. E. Goldberg. *Genetic Algorithms in search, optimization and machine learning.* Addison-Wesly, 1989.

[8] C. M. Fonseca and P. J. Fleming. Genetic algorithms for multiobjective optimization: Formulation, discussion and generalization. In *Proceedings of the 5th international coference on genetic algorithms*, pp. 416–423, 1993.

[9] E. Zitzler and L. Thiele. Multiobjective Evolutionary Algorithms: A Comparative Case Study and the Strength Pareto Approach. *IEEE Transactions on Evolutionary Computation*, Vol. 3, No. 4, pp. 257–271, 1999.

[10] K. Deb, S. Agarwal, A. Pratap, and T. Meyarivan. A Fast Elitist Non-Dominated Sorting Genetic Algorithm for Multi-Objective Optimization: NSGA-II. In *KanGAL report 200001, Indian Institute of Technology, Kanpur, India*, 2000.

[11] E. Zitzler, M. Laumanns, and L. Thiele. SPEA2: Improving the Performance of the Strength Pareto Evolutionary Algorithm. In *Technical Report 103, Computer Engineering and Communication Networks Lab (TIK), Swiss Federal Institute of Technology (ETH) Zurich*, 2001.

[12] Tamaki Okuda, Tomoyuki Hiroyasu, Mitsunori Miki, and Shinya Watanabe. DCMOGA: Distributed Cooperation model of Multi-Objective Genetic Algorithm. In *Advances in Nature-Inspired Computation: The PPSN VII Workshops*, pp. 25–26, 2002.

[13] Kalyanmoy Deb and J. Sundar. Reference Point Based Multi-Objective Optimization Using Evolutionary Algorithms. In *GECCO '06: Proceedings of the 8th annual conference on Genetic and evolutionary computation*, pp. 635–642, 2006.

[14] R. Tanese. Distributed Genetic Algorithms. In *Proc. 3rd ICGA*, pp. 434–439, 1989.

[15] Antonio Lopez Jaimes and Carlos A. Coello Coello. MRMOGA: Parallel Evolutionary Multiobjective Optimization using Multiple Resolutions. In *in 2005 IEEE Congress on Evolutionary Computation (CEC'2005)*, pp. 2294–2301, 2005.

[16] Frank Kursawe. A Variant of Evolution Strategies for Vector Optimization. In *Parallel Problem Solving from Nature. 1st Workshop, PPSN I*, pp. 193–197, 1991.

[17] H. Sato, H. Aguirre, and K. Tanaka. Local Dominance Using Polar Coordinates to Enhance Multi-objective Evolutionary Algorithms. In *Proc. 2004 IEEE Congress on Evolutionary Computation*, pp. 188–195, 2004.

[18] J. Knowles, L. Thiele, and E. Zitzler. A Tutorial on the Performance Assessment of Stochastic Multiobjective Optimizers. In *TIK Report 214, Computer Engineering and Networks Laboratory (TIK), ETH Zurich*, 2006.

[19] H. Ishibuchi and Y. Shibata. Mating Scheme for Controlling the Diversity-Convergence Balance for Multiobjective Optimization. In *Proc. of 2004 Genetic and Evolutionary Computation Conference*, pp. 1259–1271, 2004.

[20] A. Asuncion and D. J. Newman. *UCI Machine Learning Repository. Irvine, CA: University of California, School of Information and Computer Science*, 2007. http://www.ics.uci.edu/ mlearn/MLRepository.html.

[21] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at http://www.csie.ntu.edu.tw/ cjlin/libsvm.