# Journal of Applied Sciences

ISSN 1812-5654

science alert

ANSInet
an open access publisher
http://ansinet.com

Journal of Applied Sciences 13 (18): 3705-3711, 2013
ISSN 1812-5654 / DOI: 10.3923/jas.2013.3705.3711
© 2013 Asian Network for Scientific Information

# Using Multi-objective Optimization PSO in SVM for Fingerprint Recognition

Ching-Tang Hsieh and Chia-Shing Hu
Department of Electrical Engineering, Tamkang University Taipei County, Taiwan, Republic of China

**Abstract:** The problem of fingerprint classification is discussed for many years. Support Vector Machine (SVM) is a traditional artificial intelligence algorithm developed for dealing classification problems. In this study, have used the core idea of multi-objective optimization to transform SVM into a new form. This form of SVM could help to solve the situation: In tradition, SVM is usually a single optimization equation andparameters for this algorithm can only be determined by user's experience, such as penalty parameter. Therefore, this algorithm is developed to help user prevent from suffering to use this algorithm in the above condition. It is has successfully proved that user do not need to make experiment to determine the penalty parameter C. NIST-4 database is used to assess the proposed algorithm. The experiment results show the method can get good classification results.

**Key words:** MOPSO-CD, SVM, fingerprint recognition

## INTRODUCTION

Pattern recognition has been a frequently encountered problem with a wide range of application such as fingerprint, face, voice and so on. The problem can be summarized to a decision-making process to distinguish if the test sample is complied with the criteria made by the database. Generally, the classification problem can be categorized as binary classification problems (two-class classification) and multiclass classification problems (Li, 2003). Nowadays, binary classification problem can be solved by many algorithms. For instance, neutral networks, Naïve Bayes classifier, C4.5 decision tree andalso Support Vector Machine (SVM). Algorithms mentioned above can be available in multiclass problems.

Multi-objective optimization idea was arouse recent years (Raquel and Naval, 2005). This kind of algorithms is developed on a core idea of Pareto frontier. Before this idea was known for people its can only handle the optimization problem as a single objective problem as following:

$$\begin{aligned}&\text{Min } F(x)\\&\text{subject to } g_i(x) \le 0,\ i = 1, 2, \ldots, m\\&h_i(x) = 0, \quad i = 1, 2, \ldots, p\end{aligned} \tag{1}$$

where, x is called decision variables, F(x) is objective function and $g_i(x)$ and $h_i(x)$ are inequalities and equalities constraints. Note that m and p is the number constraints. However, in realistic application its often have at least two or more objectives which are not only interacting but probably conflicting. Generally, a multi-objective optimization problem can be expressed as:

$$\begin{aligned}&\text{Min } \bar{f}(x) = [f_1(x), f_2(x), \ldots, f_k(x)]\\&\text{subject to } g_i(x) \le 0,\ i = 1, 2, \ldots, m\\&h_i(x) = 0, \quad i = 1, 2, \ldots, p\end{aligned} \tag{2}$$

The desired solution for multi-objective problems is in the form of "trade-off" or compromise among the parameters that would optimize the given objectives.

This study proposed an evolutionary algorithm to reform the traditional SVM algorithm from a single objective optimization problem to a multi-objective optimization problem. The results show that can succeed to get a feasible solution without knowing penalty coefficient C and this algorithm is employed to classify fingerprint classification

**Optimization algorithm and svm:** This section presents the essential theory in this study. The definition, operations andalgorithms for multi-objective optimization algorithm and SVM are introduced as follows.

## MULTI-OBJECTIVE OPTIMIZATION

**Methodolgy:** The definition of this problem can be seen from (1). Multi-Objective Optimization (MOO) deals with generating the Pareto frontier. It is the set of non-dominated solutions for problems having more than

**Corresponding Author:** Ching-Tang Hsieh, Department of Electrical Engineering, Tamkang University Taipei County, Taiwan, Republic of China

one objective. The following definitions are shown in (Raquel and Naval, 2005).

**Definition 1:** Assume there are two solutions $x_1$ and $x_2$, if both of them are complied with the following rule its said $x_1$ dominates $x_2$:

$$\begin{aligned} &\forall i \in \{1, 2, ...., N\},\ F_i(x_1) \le F_i(x_2) \\ &\exists j \in \{1, 2, ...., N\},\ F_j(x_1) \le F_j(x_2) \end{aligned} \tag{3}$$

$x_1$ is non-dominated solution while $x_2$ is so-called dominated solution.

**Definition 2:** A vector of decision variables $\vec{x}^* \in F \subset \Re^n$ is nondominated with respect to $\chi$, if there does not exist another $\vec{x}' \in \chi$ such that $f(\vec{x}') \prec f(\vec{x})$.

Figure 1 shows the relation of dominated and non-dominated sets. In this Fig. f1 and f2 are values for the two objective functions. This example figure is a model that both f1 and f2 functions are required to be minimized. The dominated solution is the green points (i.e., $x_1$) and the non-dominated solution is the blue points (i.e.,$x_2$). With definition 1 it is shown that f1 and f2 for the non-dominated solution are either less or equal to each of dominated solutions.

**Definition 3:** A vector of decision variables $\vec{x}^* \in F \subset \Re^n$ is Pareto-optimal if it is nondominated with respect to F, where F is the feasible region. The Pareto optimal set is defined by:

$$P^* = \{\vec{x} \in \phi \mid \vec{x} \text{ is Pareto -Optimal}\}$$

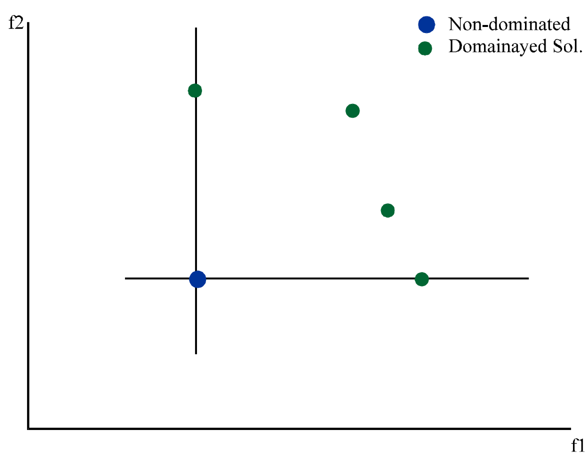


Fig. 1: The location of non-dominated solution and the relationship for a nondominated solutions with dominated solutions

**Definition 4:** The Pareto Front PF* is defined by:

$$PF^* = \{\vec{f}(\vec{x}) \in \mathbb{R}^k \mid \vec{x} \in P^*\}$$

In MOO problem, to get the Pareto optimal set from feasible set F of all the decision variables vectors satisfied constraints in (2). As noted by Margarita (Reyes-Sierra and Coello, 2006), not all the Pareto optimal set is normally desirable or achievable.

Recently, there are more and more evolutionary algorithms (EAs) have been developed in solving MOO problems, such as NSGA-II (Deb *et al*., 2000), PAES (Knowles and Corne, 2000) and SPEA2 (Van Veldhuizen and Lamont, 1998). These are all population-based algorithms which allow them to probe the different parts of the Pareto front simultaneously.

Particle Swarm Optimization (PSO) was first introduced by Kennedy and Eberhart (Kennedy and Eberhart, 1995). It is an algorithm inspired by social behavior of bird flocking. In this algorithm it will randomly distribute the population of particles in the search space. For every generation, each particle will move toward the Pareto front by the formula of updating velocity andthe best solution for a particle has achieved so far and follows the best solutions achieved among all the population particles.

Among those EAs that extend PSO to solve MOO problems is Multi-objective Particle Swarm Optimization (MOPSO) (Coello *et al*., 2004), the aggregating function for PSO (Parsopoulos and Vrahatis, 2002), or Non-dominated Sorting Particle Swarm Optimization (NSPSO) (Li, 2003).

Figure 2 shows this pseudocode for MOPSO, this figure shows the optimization algorithm structure.

The MOPSO algorithm that adopt is from (Raquel and Naval, 2005) which used the concept of Crowding Distance (CD) andit is called MOPSO-CD algorithm. Note that the formula of updating velocity is stated as:

```
Begin
Intiliazition swarm, velocities and best position
Intialize external archive (usually empty)
While
For each particle
Select a member of the external archive (if needed)
Update velocity and position
Evalute new position
Update best position and external archive
End for
```

Fig. 2: Structure of MOPSO-a simple pseudo-code

$$V_i = W \times V_i + R_1 \times (PBEST_i - P_i) + R_2 \times (A_{GBEST} - P_i) \quad (4)$$

Variables in (4) are:

| | | |
|---|---|---|
| Vi | : | Velocity |
| w | : | Intertia |
| R1, R2 | : | Random number betweenm 0 to 1 |
| $P_i$ | : | The I-th particle |
| $PBEST_i$ | : | The I-th particle personal best soulution |
| $A_{GBEST}$ | : | The pareto best solution in archive |

It's used the following test problems to verify the implementation is correct.

**Multi-objective SVM:** That traditional SVM are not able to optimize for non-positive semi definite kernel functions. In this study its use the formulation from previous research (Mierswa, 2007). For the reason it can control the over-fitting and omit the penalty factor C, where the two objective functions: maximizing margin and minimizing the number of training errors:

$$V_i = \text{Maximize Margin}: \begin{cases} \text{minimize} \frac{1}{2}\|w\|^2 \\ \text{subject to } \forall i: y_i(<w, x_i> + b) \geq 1 - \xi_i \\ \forall i: \xi \geq 0 \end{cases} \quad (5)$$

and:

$$\text{Minimze Training Error}: \begin{cases} \text{minimize} \sum_{i=2}^{n} \xi_i \\ \text{subject to } \forall i: y_i(<w, x_i> + b) \geq 1 - \xi_i \\ \forall i: \xi \geq 0 \end{cases} \quad (6)$$

After getting the two objectives, can transform both of them into dual forms:

$$\begin{aligned} &\text{Maximize} \sum_{i=1}^{n} \alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \\ &\text{subject to } \alpha_i \geq 0 \quad \text{for } i = 1, 2, \ldots, n \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \end{aligned} \quad (7)$$

$$\text{Maximize} \sum_{i=1}^{n} \alpha_i \text{ subject to } \alpha_i \geq 0 \quad \text{for } i = 1, 2, \ldots, n \text{ and } \sum_{i=1}^{n} \alpha_i y_i = 0 \quad (8)$$

Note that $k(x_i, x_j)$ is the kernel function andin this study is used Radial Basis Function (RBF) as kernel function which expression is:

$$k(x_i, x_j) = \exp\left(-\frac{\| x_i - x_j \|^2}{2\sigma^2}\right)$$

andused grid search algorithms to determine parameter in RBF.

The kernel function $k(x_i, x_j)$ can be expressed as product of $\Phi(x_i)$ $\Phi$ $(x_j)$. For non-linear classification problem, the classifier can be reformed as the following Equation:

$$f(x, b) = \text{sign}\left(\sum_{i=1}^{n} \alpha_i y_i k(x, x_i) + b\right) \quad (9)$$

Since, both of (7) and (8) share a common term:

$$\sum_{i=1}^{n} \alpha_i$$

this part of the first objective functions is not conflicting with the second one in general. Its can just omit this term in (7).

By formulating the problem b = 0, all solution hyperplanes will contain the origin and the constraints:

$$\sum_{i=1}^{n} \alpha_i y_i = 0$$

will just vanish (Burges, 1998). But if to want the equality constraint to be fulfilled it can simply be defined a third objective function:

$$-\left|\sum_{i=1}^{n} \alpha_i y_i\right|$$

Thus the problem can be reformed as:

$$\text{Maximize} \begin{cases} -\frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \\ \sum_{i=1}^{n} \alpha_i \\ -\left|\sum_{i=1}^{n} \alpha_i y_i\right| \end{cases} \quad (10)$$

By solving (10), can get the Pareto frontier, yet to still cannot decide which solution on the Pareto frontier is what to need. In previous research, have two ways to decide the solution: Maximum margin model and prediction. Two ways to search the final solution are stated as following:

**Maximum margin model:** Calculate (11) for particles on Pareto frontier and choose the one with biggest value for result of (11):

$$\sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j k(x_i, x_j) \tag{11}$$

**Minimum prediction error:** Calculate (11) for particles on Pareto frontier and choose the one with biggest value for result of (11):

$$l(y, f(x)) = \begin{cases} 1 & \text{if} \quad y \neq f(x) \\ 0 & \text{otherwise} \end{cases} \quad Err_p = \sum_{q=1}^{n} l(y_p, f_p(x_q)) \tag{12}$$

The variable k is for a small hold-out set of the data points of size. These k data p oints were part of the input training set and are not used by the learner during optimization process. After finish optimization, the learner is applied to all k data points. l(y, f(x)) is just the binary loss andfor p-th particle the errors is $E_{rrp}$. Just plot all errors $E_{rrp}$ and compare it with the original Pareto front andchoose the place where the training error and the generalization error are close to each other. This way was proved to control over fitting.

## RESULTS

In this study, there is no research to suggest the appropriate value of k in (12), can try another way to determine the error: Use all of training set for MOO process and k is equal to the number of training set samples. In this research it is proved to be useful in determine the final solution too.

To check if this implementation is feasible its use 2,000 randomly distributed samples in Fig. 3.

Figure 3 is an example that 2,000 samples with two variables ($x_1$, $x_2$), this figure just shows two sets can be separated by a linear classifier. This figure shows the distribution of the two sets (red and blue points). This is a simple example but can be easily verified the algorithm in this study.

Note that in the original research (Gao and Er, 2003), the author proposed another way to determine solution from the Pareto frontier which can be shown as following:

- Separate the training set, take out about 20% set from the training set as a test set
- Follow (12) to calculate the prediction error with the result of training error to form another plot andthen user can determine the solution they desired to avoid over-fitting for SVM result

Figure 4 is the Pareto frontier for Fig. 3. The x-axis is the objective function for Training error in the second equation from (3) and the y-axis shows the value of Margin size in the first equation from (3).

Compared with the method to took from Fig. 5-6 its simply found that can get the same result for Fig. 3 which proved this way is feasible and in some cases, more intuitive.

Figure 7 is the way proposed from the original research (Gao and Er, 2003). It transferred the training error from Fig. 4 and normalized it with total number of Pareto-optimal solutions. As the original authors pointed, this figure can help users to choose solutions without over-fitting for the solution where the training error and prediction error are the closest in the figure. This study proposed the way to choose classifier in Fig. 5 and 6 because in real application its often just need the smallest prediction error as a

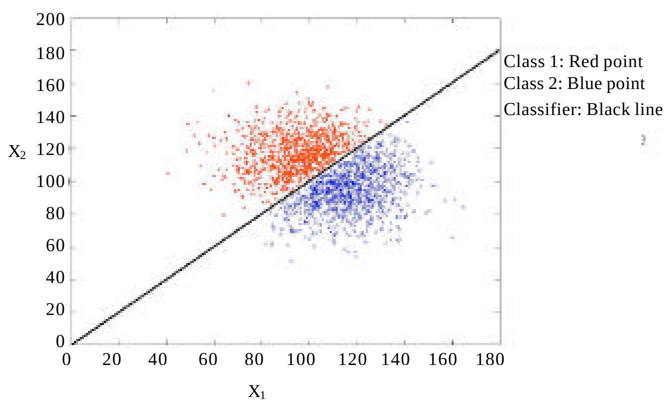


Fig. 3: Samples test if the classifier is feasible: The horizontal axis $x_1$ and vertical axis $x_2$ are the test sample which is composed of ($x_1$, $x_2$)

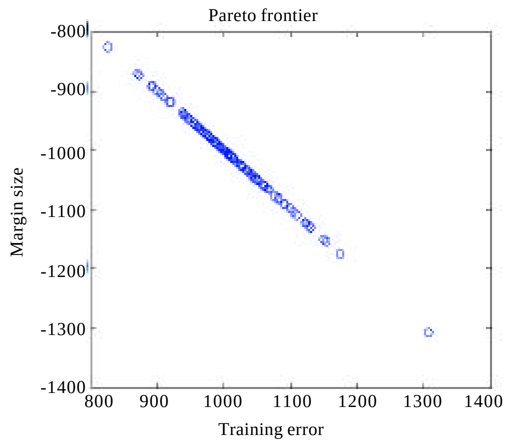


Fig. 4: Pareto frontier for samples results in Fig. 3. The vertical axis is margin size calculated by (11) and the horizontal axis is training error which is calculated by (12)

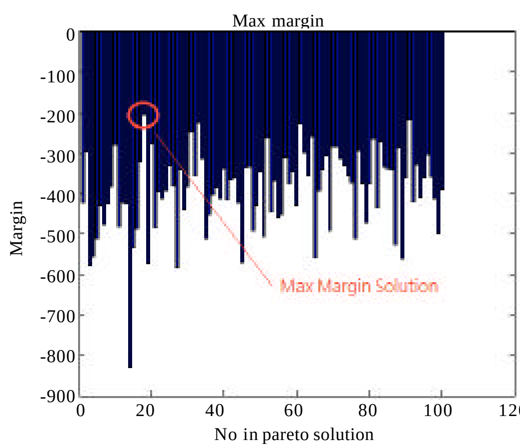


Fig. 5: Maximum margin for pareto solutions in Fig. 4. The vertical axis value is the same as Fig. 4 while the horizontal axis is every solution in Fig. 4

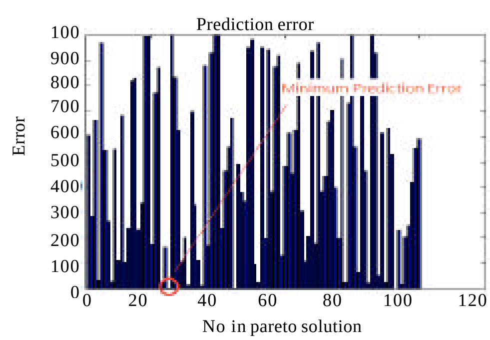


Fig. 6: Prediction Error (in this research) for pareto solutions in Fig. 4. The vertical axis value is the same as training error in Fig. 4 while the horizontal axis is every solution in Fig. 4

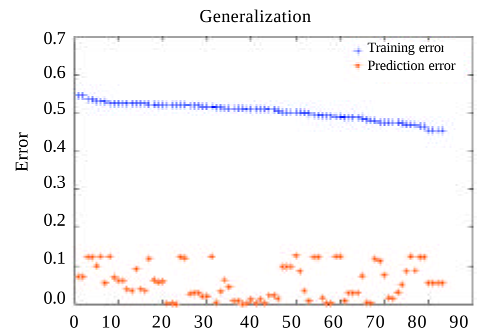


Fig. 7: Prediction error and training error comparison: The y-axis denotes the prediction error for the training(+) and testing(*) data andthe x-axis denotes a counter over all Pareto-optimal solutions ordered by training errors. Note that all the error are generalized

pointer and this is more intuitive for user to understand the rule to choose suitable Pareto solution.

In this example, the generate 100 particles, 100 generations, 100 capacities for extern archive, probability 50% for mutation and 1.4 for inertia w. By using result from Fig. 6 its still can find a good classifier to 100% distinguish two classes. Note that in this example is used the grid search algorithm to determine the σ of RBF as 0.001.

## FUZZY IMAGE ENCODER

Fuzzy logic provides human reasoning capabilities to capture uncertainties. That cannot be described by precise mathematical models (Mierswa, 2007). And fuzzy logic can able to the reasoning with some particular form of knowledge (Sagar *et al*., 1995).

Pattern identification is essentially the search for “the structure” in data andfuzzy logic is able to model the vagueness of “the structure”. There is an intimate relationship between the theory of fuzzy logic and the theory of pattern identification. The relationship is made stronger by the fact that fingerprint patterns are fuzzy in nature (Ghassemian, 1996).

In a rule-based fuzzy system to inspect fingerprint, typical rules may be:

- IF the bifurcations are PLENTY in the UPPER-RIGHT CORNER THEN the user id is Alex
- IF the bifurcations are PLENTY in the LOWER-RIGHT CORNER THEN the user id is Bob
- IF the bifurcations are PLENTY in the UPPER-RIGHT CORNER AND the bifurcations are THIN in the LOWER-RIGHT CORNER THEN the user id is Charles

- Therefore a "fuzzy feature image" encoder is applied for representing "the structure" of bifurcation point features extracted from fingerprints. The fuzzy encoder is a kind of transformation from crisp set to fuzzy set

The fuzzy encoder consists of three main steps:

- First of all, a 512x512 fingerprint image is segmented into 8×8 grids andthe width of each grid is 64 pixels as shown in Fig. 8. A fuzzy set is associated with each grid region which is shown in Fig. 9. It's use cone membership function to design the fuzzy encoder. The process of the fuzzy encoder is described as the following three steps
- In the second step a membership value is considered for each fingerprint bifurcation, wherein a cone membership function is performed for each grid in order to present the structure of bifurcation features (Fig. 9). The results of this analysis are used to get the membership value of the bifurcation to the fuzzy sets considered in previous step. The membership function of grid (i, j) is computed as:

$$\mu(i,j)=\sum_{n=1}^{m}\left(1-\frac{\text{DistanceToGridCenter}_n}{\text{GridWidth}}\right) \tag{13}$$

where, μ (i, j) is the membership function of grid (i, j), m is the number of bifurcation points near the center of grid (i, j) andthe Grid Width in this study is 64.

Finally, calculate the sum of membership degrees in each grid. Then the fuzzy image of fingerprint bifurcation structure is obtained in the third step.

The gray level value of fuzzy image is computed as:

$$F(i,j)=\begin{cases}255 & \text{if } \mu(i,j)\geq 1\\ \mu(i,j)\times 255 & \text{if } 0\leq\mu(i,j)<1\\ 0 & \text{if } \mu(i,j)<0\end{cases} \tag{14}$$

where, F (i, j) is the gray level value of grid (i, j) in a fuzzy image which is shown in Fig. 10.

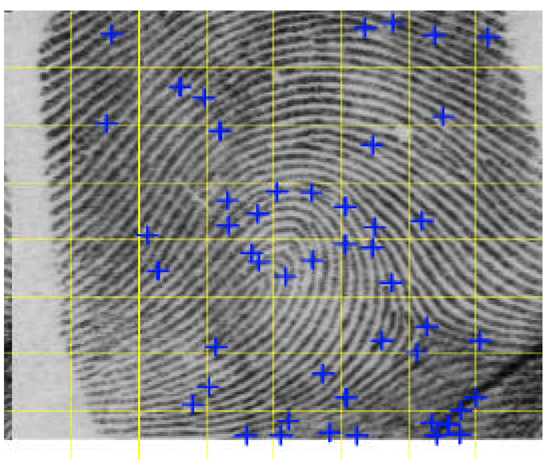

Fig. 8: A sample image with the bifurcation points in 8×8 grids

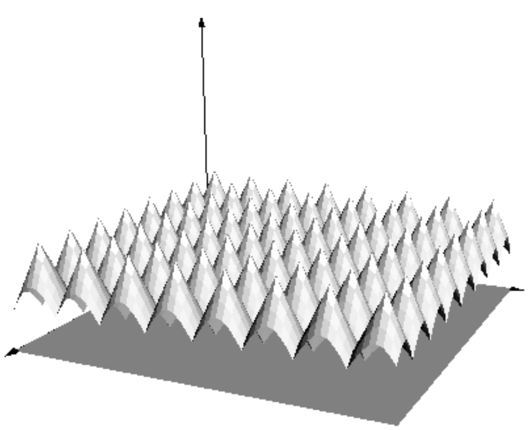

Fig. 9: Membership functions of the fuzzy encoder

## EXPERIMENTAL RESULTS

In this study, has proposed a new algorithm to transform traditional SVM into multi-objective dual form and use MOPSO-CD algorithm to solve the dual problem.

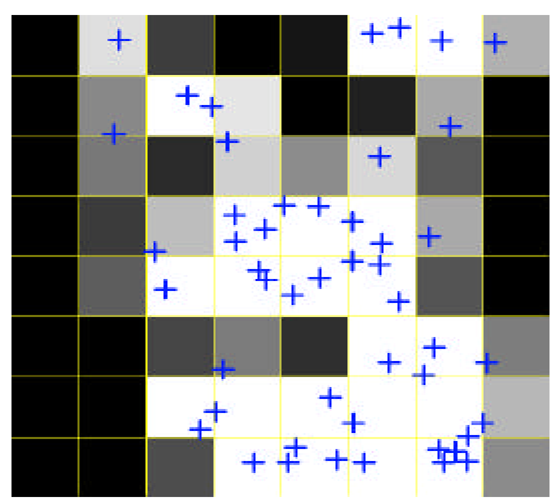

Fig. 10: The fuzzy image of fingerprint bifurcation structure

To train the dataset in NIST-4 into this algorithm, take the fuzzy encoder into a 1x64 array to represent this image sample.

The way proposed in this research to choose suitable Pareto solution in Pareto frontier is lightly different from the original research (Gao and Er, 2003). The criteria in this study is more intuitive than the original way. The original way is used to choose solution with the smallest over-fitting condition, yet the proposed way in this study focused on the intuitive way for user to determine the solution. The former way requires users to own more knowledge of SVM and Multi-objective optimization, yet the way in this research only requires users to know to choose the solution with the minimum training error or max margin it should be more intuitive and comply with the central idea of this study-to provide researchers the more easier way to use SVM without trial and error on determining the penalty coefficient C.

## CONCLUSION

In this study, has developed a multi-objective optimized SVM algorithm which is proved effective for binary-class fingerprint classification. This algorithm can reduce the work from manually operation for testing suit parameters of SVM. Note that even if without this algorithm, people still can spend lots of time building figures in this research. The suggested a more logical and more time-effective way to evaluate the proper SVM parameters compared to other literatures.

However, that did not apply the algorithm to multi-class labeled problem. As a future work, has should put this algorithm forward to applying to multi-class problem.

## REFERENCES


Burges, C.J.C., 1998. A tutorial on support vector machines for patter recognition. Data Mining Knowledge Discovery, 2: 121-167.

Coello, C.A.C., G.T. Pulido and M.S. Lechuga, 2004. Handling multiple objectives with particle swarm optimization. IEEE Tran. Evol. Comput., 8: 256-279.

Deb, K., S. Agrawal, A. Pratap and T. Meyarivan, 2000. A fast elitist non-dominated sorting genetic algorithm for multi-objective optimization: NSGA-II, Parallel Probl. Solving Nat., 1917: 849-858.

Gao, Y. and M.J. Er, 2003. Online adaptive fuzzy neural identification and control of a class of MIMO nonlinear systems. IEEE Trans. Fuzzy Syst., 11: 462-477.

Ghassemian, M.H., 1996. A robust on-line restoration algorithm for fingerprint segmentation. Proceedings of the International Conference on Image Processing, Volume 1, September 16-19, 1996, Lausanne, Switzerland, pp: 181-184.

Kennedy, J. and R. Eberhart, 1995. Particle swarm optimization. Proceedings of the International Conference on Neural Networks, Volume 4, November 27-December 1, 1995, Perth, WA., USA., pp: 1942-1948.

Knowles, J.D. and D.W. Corne, 2000. Approximating the nondominated front using the Pareto archived evolution strategy. Evol. Comput., 8: 149-172.

Li, X., 2003. A non-dominated sorting particle swarm optimizer for multiobjective optimization. Proceedings of the Genetic and Evolutionary Computation Conference, July 12-16, 2003, Chicago, IL., USA., pp: 37-48.

Mierswa, I., 2007. Controlling overfitting with multi-objective support vector machine. Proceedings of the 9th Annual Conference on Genetic and Evolutionary Computation, July 7-11, 2007, London, UK., pp: 1830-1837.

Parsopoulos, K.E. and M.N. Vrahatis, 2002. Particle swarm optimization method in multiobjective problems. Proceedings of the ACM Symposium on Applied Computing, March 11-14, 2002, Madrid, Spain, pp: 603-607.

Raquel, C.R. and P.C. Naval, 2005. An effective use of crowding distance in multiobjective particle swarm optimization. Proceedings of the Conference on Genetic and Evolutionary Computation, Jun. 25-29, ACM, New York, USA., pp: 257-264.

Reyes-Sierra, M. and C.A.C. Coello, 2006. Multi-Objective particle swarm optimizers: A survey of the state-of-the-art. Int. J. Comput. Intell. Res., 2: 287-308.

Sagar, V.K., D.B.L. Ngo and K.C.K. Foo, 1995. Fuzzy feature selection for fingerprint identification. Proceedings of the IEEE 29th Annual International Carnahan Conference on Security Technology, October 18-20, 1995, Sanderstead, UK., pp: 85-90.

Van Veldhuizen, D. and G. Lamont, 1998. Multiobjective evolutionary algorithms research: A history and analysis. Technical Report TR-98-03, Air Force Institute of Technology, USA.