# 行政院國家科學委員會專題研究計畫　期中進度報告

## 子計畫一：行動電子商務中多策略學習之分散式智慧型代理人架構(2/3)

計畫類別： 整合型計畫
計畫編號： NSC91-2213-E-002-044-
執行期間： 91 年 08 月 01 日至 92 年 07 月 31 日
執行單位： 國立臺灣大學電機工程學系暨研究所


計畫主持人： 王勝德

計畫參與人員： 林俊甫 吳國賓 陳叡立 郭文雄


報告類型： 精簡報告

處理方式： 本計畫可公開查詢

中　華　民　國 92 年 6 月 2 日

計畫名稱：子計畫一：行動電子商務中多策略學習之分散式智慧型代理人架構

(2/3)

計畫主持人 王勝德

計畫參與人員：林俊甫 吳國賓 陳叡立 郭文雄

期中報告

計畫編號：91-2213-E-002-044-

# 中文摘要：

關鍵詞：分類，機器學習，雜訊，模糊成員函數

學習演算法(Learning Algorithms)是智慧型代理人(Intelligent Agent)中最基本的組成部分，藉由不同的學習演算法，可使智慧型代理人表現出不同的效能和行為。在眾多學習演算法中，分類演算法可使智慧型代理人從已知資料中學習分類的方法，進而從未來的資料中做出正確的預測或表現合適的行為。模糊向量支持機器(Fuzzy support vector machines, FSVMs)將每一個資料點聯結一個可表示資料意義的模糊成員函數，降低雜訊在學習過程的影響。在這篇報告中，我們提出及比較兩種自動設定資料點模糊成員函數的方法，使得我們能更方便地處理有雜訊的分類問題及應用模糊向量支持機器於智慧型代理人中。使用指標性的資料庫與其他演算法做比較，實驗結果證明我們的演算法可以有效的處理這個問題。

# 英文摘要：

Keywords: classification, machine learning, noise, fuzzy membership,

Learning algorithms are the basic part of intelligent agent. Different learning algorithms can affect the performance and behavior of the intelligent agent. Classification algorithms used in many intelligent agent applications are one of these learning algorithms. Intelligent agents can learn the behavior from known data and predict or behave from future data by classification algorithms. Fuzzy support vector machines (FSVMs) provide a way to classify data with noises or outliers. Each data point is associated with a fuzzy membership that can reflect their relative degrees as meaningful data. In this report, we investigate and compare two strategies of automatically setting the fuzzy memberships of data points. It makes the usage of

FSVMs easier in the application of reducing the effects of noises or outliers such that makes better behavior of intelligent agents. The experiments show that the generalization error of FSVMs is comparable to other methods on benchmark datasets.

## 報告內容：

## 1. Introduction

The theory of support vector machines (SVMs), that is based on the idea of structural risk minimization (SRM), is a new classification technique and has drawn much attention on this topic in recent years [3, 4, 9, 10]. The good generalization ability of SVMs is achieved by finding a large margin between two classes [1, 8]. In many applications, the theory of SVMs has been shown to provide higher performance than traditional learning machines [3] and has been introduced as powerful tools for solving classification problems.

Since the optimal hyperplane obtained by the SVM depends on only a small part of the data points, it may become sensitive to noises or outliers in the training set [2, 13]. To solve this problem, one approach is to do some preprocessing on training data to remove noises or outliers, and then use the remaining set to learn the decision function. This method is hard to implement if we do not have enough knowledge about noises or outliers. In many real world applications, we are given a set of training data without knowledge about noises or outliers. There are some risks to remove the meaningful data points as noises or outliers.

There are many discussions in this topic and some of them show good performance. The theory of Leave-One-Out SVMs [11] (LOO-SVMs) is a modified version of SVMs. This approach differs from classical SVMs in that it is based on the maximization of the margin, but minimizes the expression given by the bound in an attempt to minimize the leave-one-out error. No free parameter makes this algorithm easy to use, but it lacks the flexibility of tuning the relative degree of outliers as meaningful data points. Its generalization, the theory of Adaptive Margin SVMs (AM-SVMs) [12], uses a parameter $\lambda$ to adjust the margin for a given learning problem. It improves the flexibility of LOO-SVMs and shows better performance. The experiments in both of them show the robustness against outliers.

FSVMs solve this kind of problems by introducing the fuzzy memberships of data points. The main advantage of FSVMs is that we can associate a fuzzy membership to each data point such that different data points can have different effects in the learning of the separating hyperplane. We can treat the noises or outliers as less importance and let these points have lower fuzzy membership. It is also based on the maximization of the margin like the classical SVMs, but uses fuzzy memberships to prevent some points from making narrower margin. This equips FSVMs with the ability to train data with noises or outliers by setting lower fuzzy memberships to the data points that are considered as noises or outliers with higher probability.

The previous work of FSVMs [6] did not address the issue of automatically setting the fuzzy membership from the data set. We need to assume a noise model of the training data points, and then try and tune the fuzzy membership of each data point in the training. Without any knowledge of the distribution of data points, it is hard to associate the fuzzy membership to the data point.

In this report, we propose two strategies to estimate the probability that the data point is considered as noisy information and use this probability to tune the fuzzy membership in FSVMs. This simplifies the use of FSVMs in the training of data points with noises or outliers. The experiments show that the generalization error of FSVMs is comparable to other methods on benchmark datasets.

## 2. Fuzzy Support Vector Machines

Suppose we are given a set $S$ of labeled training points with associated fuzzy memberships $(y_1, x_1, s_1), \ldots, (y_l, x_l, s_l)$. Each training point $x_i \in R^N$ is given a label $y_i \in \{-1,1\}$ and a fuzzy membership $\sigma \le s_i \le 1$ with $i$=1,...,$l$, and sufficient small $\sigma > 0$, since the data point with $s_i = 0$ means nothing and can be just removed from training set without affecting the result of optimization. Let $z = \varphi(x)$ denote the corresponding feature space vector with a mapping $\varphi$ from $R^N$ to a feature space $Z$.

Since the fuzzy membership $s_i$ is the attitude of the corresponding point $x_i$ toward one class and the parameter $\xi_i$ can be viewed as a measure of error in the SVM, the term $s_i \xi_i$ is then a measure of error with different weighting. The optimal hyperplane problem is then regarded as the solution to minimize $\frac{1}{2} w \cdot w + C \sum_{i=1}^{l} s_i \xi_i$,

subject to $\begin{cases} y_i (w \cdot z_i + b) \ge 1 - \xi_i, i = 1, \ldots, l, \\ \xi_i \ge 0, i = 1, \ldots, l, \end{cases}$ where $C$ is a constant. It is noted that a smaller $s_i$ reduces the effect of the parameter $\xi_i$ such that the corresponding point $x_i$ is treated as less important.

The only free parameter $C$ in SVMs controls the trade-off between the maximization of margin and the amount of misclassifications. A larger *C* makes the training of SVMs less misclassifications and narrower margin. The decrease of *C* makes SVMs ignore more training points and get a wider margin.

In FSVMs, we can set *C* to be a sufficient large value. It is the same as SVMs that the system will get narrower margin and allow less miscalssifications if we set all $s_i = 1$. With different value of $s_i$, we can control the trade-off of the respective training point $x_i$ in the system. A smaller value of $s_i$ makes the corresponding point $x_i$ less important in the training.

There is only one free parameter in SVMs while the number of free parameters in FSVMs is equivalent to the number of training points.

## 3. Training Procedures

There are many methods to training data using FSVMs, depending on how much information contains in the data set. If the data points are already associated with the fuzzy memberships, we can just use this information in training FSVMs. If it is given a noise distribution model of the data set, we can set the fuzzy membership as the probability of the data point that is not a noise, or as a function of it. Let $p_i$ be the probability of the data point $x_i$ that is not a noise. If there exists this kind of information in the training data, we can just assign the value $s_i = p_i$ or $s_i = f(p_i)$ as the fuzzy membership of each data point. Since almost all applications lack this information, we need some other methods to predict this probability. In order to reduce the effects of noisy data when using FSVMs in this kind of problem, we propose the following training procedure.

1. Use the original algorithm of SVMs to get the optimal kernel parameters and the regularization parameter *C*.
2. Use some strategies to set the fuzzy memberships of data points and find the modified hyperplane by FSVMs in the same kernel space.

As for steps, we propose two strategies: one is based on kernel-target alignment and the other is using k-NN.

### 3.1 Strategy of Using Kernel-Target Alignment

The idea of kernel-target alignment is introduced in [5]. Let $f_K(x_i, y_i) = \sum_{j=1}^{l} y_i y_j K(x_i, x_j)$ . The kernel-target alignment is defined as

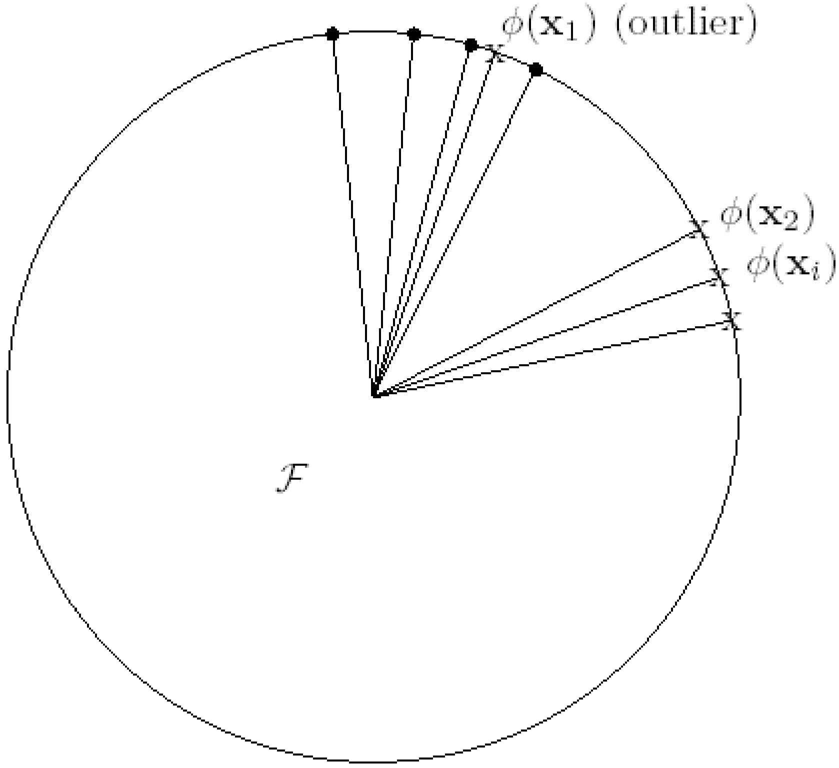


Figure 1: The value $f_K(\mathbf{x}_1, y_1)$ is lower than $f_K(\mathbf{x}_2, y_2)$ in the RBF kernel.

$A_{KT} = \dfrac{\sum_{i=1}^{l} f_K(x_i, y_i)}{l\sqrt{\sum_{i=1}^{l} K^2(x_i, x_j)}}$ . This definition provides a method for selecting kernel parameters and the experimental results show that adapting the kernel to improve alignment on the training data enhances the alignment on the test data, thus improved classification accuracy.

In order to discover some relation between the fuzzy membership and the data point, we simply focus on the value $f_K(x_i, y_i)$. Suppose $K(x_{i,} x_j)$ is a kind of distance measure between data points $x_i$ and $x_j$ in feature space *Z*. For example, by using the RBF kernel $K(x_i, x_j) = e^{-\gamma\|x_i - x_j\|^2}$, the data points live on the surface of a hypersphere in feature space *Z* as shown in Figure 1. Then $K(x_i, x_j) = \varphi(x_i)\cdot\varphi(x_j)$ is the cosine of the angle between $\varphi(x_i)$ and $\varphi(x_j)$. For the outlier $\varphi(x_1)$ and the representative $\varphi(x_2)$, we can easily check the value $f_K(x_1, y_1)$ is lower than $f_K(x_2, y_2)$.

We observe this situation and assume that the data point $x_i$ with lower value of $f_K(x_i, y_i)$ can be considered as outlier and should make less contribution of the classification accuracy. For this assumption, we can build a relationship between the fuzzy membership $s_i$ and the value of $f_K(x_i, y_i)$ that is defined as

$$s_i = \begin{cases} 1, f_K(x_i, y_i) > f_K^{UB} \\ \sigma, f_K(x_i, y_i) < f_K^{LB} \\ \sigma + (1-\sigma)\left(\dfrac{f_K(x_i, y_i) - f_K^{LB}}{f_K^{UB} - f_K^{LB}}\right)^d, otherwise \end{cases}$$

, where $f_K^{UB}$ and $f_K^{LB}$ are the parameters that control the mapping region between $s_i$ and $f_K(x_i, y_i)$, and $d$ is the parameter that controls the degree of mapping function.

The training points are divided into three regions by the parameters $f_K^{UB}$ and $f_K^{LB}$. The data points in the region with $f_K(x_i, y_i) > f_K^{UB}$ can be viewed as valid examples and the fuzzy membership is equal to 1. The data points in the region with $f_K(x_i, y_i) < f_K^{LB}$ can be highly thought as noisy data and the fuzzy membership is assigned to $\sigma$. The data points in rest region are considered as noise with different probabilities and can make different distributions in the training process.

### 3.2 Strategy of Using k-NN

For each data point $x_i$, we can find a set $S_i^k$ that consists of *k* nearest neighbors of $x_i$. Let $n_i$ be the number of data points in the set $S_i^k$ that the class label is the same as the class label of data point $x_i$. It is reasonable to assume that the data point with lower value of $n_i$ is more probable as noisy data. But for the data points that are near the margin of two classes, the value $n_i$ of these points may be lower. It will get poor performance if we set these data points with lower fuzzy memberships. In order to avoid this situation, we introduce a parameter $k^{UB}$ that controls the threshold of which data point needs to reduce its fuzzy membership.
For this assumption, we can build a relationship between the fuzzy membership $s_i$ and the value of $n_i$ that is defined as $s_i = \begin{cases} 1, n_i > k^{UB} \\ \sigma + (1-\sigma)\left(\dfrac{n_i}{k^{UB}}\right)^d, otherwise \end{cases}$, where $d$ is the parameter that controls the degree of mapping function.

## 4. Experiments

We conducted computer simulations of SVMs and FSVMs using the data sets as in [7]. In these simulations, we use the RBF kernel as $K(x_i, x_j) = e^{-\gamma \|x_i - x_j\|^2}$. For each data set, we train and test the first 5 sample sets to find the best parameters and use these parameters to get the result of the whole sample sets. Since there are more parameters than the original algorithm of SVMs, we use two procedures to find the parameters as described in the previous section. In the first procedure, we search the kernel parameters and *C* using the original algorithm of SVMs. In the second procedure, we fix the kernel parameters and *C* that are found in the first stage, and search the parameters of the fuzzy membership mapping function.

To find the parameters of strategy using kernel-target alignment, we first fix $f_K^{UB} = \max_i f_K(x_i, y_i)$ and $f_K^{LB} = \min_i f_K(x_i, y_i)$, and perform a two-dimensional search of parameters $\sigma$ and *d*. The value of $\sigma$ is chosen from 0.1 to 0.9 step by 0.1. For some case, we also compare the result of $\sigma$ =0.01. The value of *d* is chosen from $2^{-8}$ to $2^8$ multiply by 2. Then, we fix $\sigma$ and *d*, and perform a two-dimensional search of parameters $f_K^{UB}$ and $f_K^{LB}$. The value of $f_K^{UB}$ is chosen such that 0%, 10%, 20%, 30%, 40%, and 50% of data points have the value of fuzzy membership as 1. The value of $f_K^{LB}$ is chosen such that 0%, 10%, 20%, 30%, 40%, and 50% of data points have the value of fuzzy membership as $\sigma$.

TABLE 1: THE PARAMETERS USED IN SVMS, FSVMS USING STRATEGY OF KERNEL-TARGET ALIGNMENT (KT), AND FSVMS USING STRATEGY OF K-NN (K-NN) ON 13 DATASETS.

| | SVMs | | KT | | | | k-NN | |
|---|---|---|---|---|---|---|---|---|
| | $C$ | $\gamma$ | $\sigma$ | $d$ | UB | LB | $\sigma$ | $k$ |
| Banana | 316.2 | 1 | 0.01 | 64 | 10% | 0% | 0.1 | 32 |
| B. Cancer | 15.19 | 0.02 | 0.5 | 8 | 20% | 0% | 0.01 | 64 |
| Diabetes | 1 | 0.05 | 0.7 | 8 | 10% | 0% | 0.6 | 4 |
| German | 3.162 | 0.01818 | 0.6 | 8 | 20% | 30% | 0.8 | 4 |
| Heart | 3.162 | 0.00833 | 0.3 | 16 | 30% | 30% | 0.2 | 32 |
| Image | 500 | 0.03333 | 0.3 | $2^{-3}$ | 10% | 0% | - | - |
| Ringnorm | 1e+9 | 0.1 | - | - | - | - | - | - |
| F. Solar | 1.023 | 0.03333 | 0.5 | $2^{-4}$ | 20% | 0% | 0.3 | 256 |
| Splice | 1000 | 0.14286 | - | - | - | - | - | - |
| Thyroid | 10 | 0.33333 | 0.7 | $2^{-6}$ | 0% | 0% | - | - |
| Titanic | 100000 | 0.5 | 0.5 | 32 | 30% | 0% | 0.2 | 128 |
| Twonorm | 3.162 | 0.025 | 0.01 | 128 | 10% | 0% | 0.01 | 128 |
| Waveform | 1 | 0.05 | 0.01 | $2^{-8}$ | 50% | 0% | - | - |

To find the parameters of strategy using k-NN, we just perform a two-dimensional search of parameters $\sigma$ and *k*. We fix the value $k^{UB}$ = 0.5*k* and *d*=1 since we don't find much improvement when we choose other values of these two parameters such that we skip searching for saving time. The value of $\sigma$ is chosen from 0.1 to 0.9 stepped by 0.1. For some case, we also compare the result of $\sigma$=0.01. The value of *k* is chosen from 2 to $2^8$ multiplied by 2. Table 1 lists the parameters after our optimization in the simulations. For some data sets, we cannot find any parameters that can improve the performance of SVMs such that we left blank in this table.

Table 2 shows the results of our simulations. For comparison with SVMs, FSVMs with kernel-target alignment perform better in 9 data sets, and FSVMs with k-NN perform better in 5 data sets. By checking the average training error of SVMs in each data set, we find that FSVMs perform well in the data set when the average training error is high. These results show that our algorithm can improve the performance of SVMs when the data set contains noisy data.

We also list in Table 3 the other results for single RBF classifier (RBF), AdaBoost (AB), and regularized AdaBoost (ABR), that are obtained from [7], and the results for LOO-SVM, that are obtained from [12]. We can easily check that FSVMs perform better in the data set with noises.

Table 2: The average training error of SVMs (TR), and the test error of SVMs, FSVMs using strategy of kernel-target alignment (KT), and FSVMs using strategy of k-NN (k-NN) on 13 datasets.

| | TR | SVMs | KT | k-NN |
|---|---|---|---|---|
| Banana | 6.7 | 11.5 | 10.4 | 11.4 |
| B. Cancer | 18.3 | 26.0 | 25.3 | 25.2 |
| Diabetes | 19.4 | 23.5 | 23.3 | 23.5 |
| German | 16.2 | 23.6 | 23.3 | 23.6 |
| Heart | 12.8 | 16.0 | 15.2 | 15.5 |
| Image | 1.3 | 3.0 | 2.9 | - |
| Ringnorm | 0.0 | 1.7 | - | - |
| F. Solar | 32.6 | 32.4 | 32.4 | 32.4 |
| Splice | 0.0 | 10.9 | - | - |
| Thyroid | 0.4 | 4.8 | 4.7 | - |
| Titanic | 19.6 | 22.4 | 22.3 | 22.3 |
| Twonorm | 0.4 | 3.0 | 2.4 | 2.9 |
| Waveform | 3.5 | 9.9 | 9.9 | - |

Table 3: Comparison of test error of Single RBF classifier, AdaBoost (AB), Reglarized AdaBoost ($AB_R$), SVMs, LOO-SVMs (LOOS), FSVMs using strategy of kernel-target alignment (KT), and FSVMs using strategy of k-NN (k-NN) on 13 datasets.

| | RBF | AB | $AB_R$ | SVMs | LOOS | KT | k-NN |
|---|---|---|---|---|---|---|---|
| Banana | 10.8 | 12.3 | 10.9 | 11.5 | 10.6 | 10.4 | 11.4 |
| B. Cancer | 27.6 | 30.4 | 26.5 | 26.0 | 26.3 | 25.3 | 25.2 |
| Diabetes | 24.3 | 26.5 | 23.8 | 23.5 | 23.4 | 23.3 | 23.5 |
| German | 24.7 | 27.5 | 24.3 | 23.6 | N/A | 23.3 | 23.6 |
| Heart | 17.6 | 20.3 | 16.5 | 16.0 | 16.1 | 15.2 | 15.5 |
| Image | 3.3 | 2.7 | 2.7 | 3.0 | N/A | 2.9 | - |
| Ringnorm | 1.7 | 1.9 | 1.6 | 1.7 | N/A | - | - |
| F. Solar | 34.4 | 35.7 | 34.2 | 32.4 | N/A | 32.4 | 32.4 |
| Splice | 10.0 | 10.1 | 9.5 | 10.9 | N/A | - | - |
| Thyroid | 4.5 | 4.4 | 4.6 | 4.8 | 5.0 | 4.7 | - |
| Titanic | 23.3 | 22.6 | 22.6 | 22.4 | 22.7 | 22.3 | 22.3 |
| Twonorm | 2.9 | 3.0 | 2.7 | 3.0 | N/A | 2.4 | 2.9 |
| Waveform | 10.7 | 10.8 | 9.8 | 9.9 | N/A | 9.9 | - |

## 5. Conclusions

In this paper, we propose training procedures for FSVMs, and describe two strategies for setting fuzzy membership in FSVMs. It makes FSVMs more feasible in the application of reducing the effects of noises or outliers. The experiments show that the performance is better in
the applications with the noisy data.

We also compare the two strategies for setting the fuzzy membership in FSVMs. The usage of FSVMs with kernel-target alignment is more complicated since there exist many parameters. It costs much time to find the optimal parameters in the training process but the performance is better. The usage of FSVMs using k-NN is much simple to use and the results are close to the previous strategy.

## 參考文獻：


[1] P. Bartlett and J. Shawe-Taylor, "Generalization Performance of Support Vector Machines and Other Pattern Classifiers," in B. Scholkopf, C. Burges and A. Smola (eds.), **Advances in Kernel Methods: Support Vector Learning**, Cambridge, MA: MIT Press, pp. 43-54, 1998.

[2] B. E. Boser, I. Guyon and V. Vapnik, "A Training Algorithm for Optimal Margin Classifiers," in **Computational Learning Theory**, 1992, pp. 144-152.

[3] C. J. C. Burges, "A Tutorial on Support Vector Machines for Pattern Recognition," **Data Mining and Knowledge Discovery**, vol. 2, no. 2, pp. 121-167, 1998.

[4] C. Cortes and V. Vapnik, "Support Vector Networks," **Machine Learning**, vol. 20, pp. 273-297, 1995.

[5] N. Cristianini, J. Shawe-Taylor, A. Elisseeff and J. Kandola, "On Kernel-Target Alignment," in **Advances in Neural Information Processing Systems 14**, MIT Press, 2002, pp. 367-373.

[6] C. F. Lin and S. D. Wang, "Fuzzy Support Vector Machines," **IEEE Transactions on Neural Networks**, vol. 13, no. 2, pp. 464-471, 2002.

[7] G. Ratsch, T. Onoda and K.-R. Muller, "Soft Margins for AdaBoost," **Machine Learning**, vol. 42, no. 3, pp. 287-320, 2001.

[8] J. Shawe-Taylor and P. L. Bartlett, "Structural Risk Minimization Over Data-dependent hierarchies," **IEEE Transactions on Information Theory**, vol. 44, no. 5, pp. 1926-1940, 1998.

[9] V. Vapnik, **The Nature of Statistical Learning Theory**, New York: Springer, 1995.

[10] V. Vapnik, **Statistical Learning Theory**, New York: Springer, 1998.

[11] J. Weston, "Leave-One-Out Support Vector Machines," In T. Dean (ed.), **Proceedings of the Sixteenth International Joint Conference on Artificial**

**Intelligence, IJCAI 99**, Morgan Kaufmann, 1999, pp. 727-733.
[12] J. Weston and R. Herbrich, “Adaptive Margin Support Vector Machines,” in A. Smola, P. Bartlett, B. Scholkopf and D. Schuurmans (eds.), **Advances in Large Margin Classifiers, Cambridge**, MA: MIT Press, pp. 281-295, 2000.
[13] X. Zhang, “Using Class-Center Vectors to Build Support Vector Machines,” in **Neural Networks for Signal Processing IX**, 1999, pp. 3-11.

## 計劃成果自評：

本計畫定期舉辦研討會，並投稿國內外學術會議，以及 IEEE Trans. Neural Networks 期刊，成效良好。