Model Selection for Support Vector Machines Using Ant Colony Optimization in an Electronic Nose Application

Javier Acevedo, Saturnino Maldonado, Sergio Lafuente, Hilario Gomez, and Pedro Gil

University of Alcala, Teoría de la señal, Alcala de Henares, Spain
javier.acevedo@uah.es

Abstract. Support vector machines, especially when using radial basis kernels, have given good results in the classification of different volatile compounds. We can achieve a feature extraction method adjusting the parameters of a modified radial basis kernel, giving more importance to those features that are important for classification proposes. However, the function that has to be minimized to find the best scaling factors is not derivable and has multiple local minima. In this work we propose to adapt the ideas of the ant colony optimization method to find an optimal value of the kernel parameters.

1 Introduction

Electronic noses are defined as an array of sensors and a pattern recognition (PARC) system [1]. Over the past years these systems have been applied to many different applications with a considerable success. One of the aspects gaining in importance in the electronic nose field is feature extraction [2]. The importance of this stage for the PARC system lies in the need to enhance those features that have more importance for classification. In fact, this method is quite similar to the way the brain processes the information in the olfactory bulb, giving more importance to those signals coming from the nose that are useful to identify the target odor. However, most of the work developed about feature extraction is done using the filter approach, that is, the data are transformed independently of the learning machine employed. This approach is straightforward but it is not as coherent as the wrapper approach, which makes the data transformation depending on the result of the classification machine.

Support vector machines (SVM) have demonstrated to be a powerful learning method [3] and its use in the electronic nose field is getting more importance [4]. In particular, best results are achieved with radial basis function (RBF) kernels [5]. When using such kernels, some hyperparameters must be tuned, but this process is usually done picking up some values and testing with an external dataset. In [6] the proposal was to use a multi-gamma kernel, where every feature had its own gamma parameter. In this way, the tuning of the hyperparameters can be used as a feature extraction with a wrapper approach. This

M. Dorigo et al. (Eds.): ANTS 2006, LNCS 4150, pp. 468–475, 2006.

tuning is usually done testing several values of the parameters and measuring a classification error, such as cross-validation. However, when the dimension of the problem is not very small, as it is usual in the electronic nose problems, the number of possible solutions makes it impossible to test all the combinations.

It has to be noted that the functions employed to test the classification error, and so to select the scaling factor values, are not derivable. As a matter of fact, these functions have multiple local minima, so the use of optimization methods based on a gradient descent is less than appropriated. Ant Colony Optimization (ACO) [7],[8] is a recent meta-heuristic search method, based in swarm intelligence, that is providing good results to solve hard combinatorial problems. In this work, we have adapted this method to search the scaling factors values that minimize the classification error.

2 SVM Classification Error

Given a problem with a set of training vectors $\boldsymbol{x_i} \in \mathrm{R}^n, i = 1, ..., l$ and a vector of labels $\boldsymbol{y} \in \mathrm{R}^l, y_i \in \{-1, 1\}$ the training of the SVM implies to solve the following optimization problem:

$$\begin{gathered} \min_{\alpha} \; W(\boldsymbol{\alpha}) = \tfrac{1}{2}\boldsymbol{\alpha}^T \boldsymbol{Q}\boldsymbol{\alpha} - \boldsymbol{e}^T\boldsymbol{\alpha} \\ \text{subject to } 0 \leq \alpha_i \leq C, i = 1, ..., l \\ \boldsymbol{y}^T\boldsymbol{\alpha} = 0 \; . \end{gathered} \tag{1}$$

where $\boldsymbol{e}$ is the unity vector, C is a regularization parameter and $Q_{ij} = y_i y_j K(\boldsymbol{x_i}, \boldsymbol{x_j})$ is a symmetric matrix, with $K(\boldsymbol{x_i}, \boldsymbol{x_j})$ being the kernel function. With certain kernels the Q matrix is positive definitive and so, the problem described in (1) has a unique solution of $\boldsymbol{\alpha}$ that can be found quickly using a gradient descent method or some other decomposition method like the one proposed in [9]. Once the solution of $\boldsymbol{\alpha}$ is found, there will be only a number of training patterns $\boldsymbol{x_i}$ with α_i different to zero. These patterns are known as support vectors. Then, for a new incoming pattern $\boldsymbol{x}$ we have a decision function:

$$f(\boldsymbol{x}) = \operatorname{sgn}\left(\sum_i \alpha_i y_i K(\boldsymbol{x}, \boldsymbol{x_i}) + b\right), \; \boldsymbol{x_i} \in \boldsymbol{V} \; . \tag{2}$$

being V the set of support vectors. As it has been mentioned, in the electronic nose best results have been achieved with a RBF kernel. In this work, the proposal is to use the following kernel:

$$K(\boldsymbol{x}, \boldsymbol{y}) = e^{-\sum_{i=1}^{n} \gamma_i (x_i - y_i)^2} \; . \tag{3}$$

Being $\gamma_i \in [0, 1]$ the scaling factors associated to each feature. These scaling factors have to be tuned to improve the performance of the SVM. Now, the question is how the classification error can be measured. Most of the works that use SVM with a simple RBF kernel use an external test set to adjust the values

that minimize the classification error. It is especially interesting the measure obtained with the leave-one-out procedure because it gives us an unbiased estimator of the classification error. However, this procedure is very expensive from a computational point of view, since it requires l trainings. It is well known that only support vectors are able to introduce error in the leave-one-out procedure. Moreover, in [10] it was demonstrated that if a support vector fulfils (4), then that support vector does not introduce error in the leave-one-out process:

$$2\alpha_i^0 R^2 - y_i f^0(x_i) < 0 \ . \tag{4}$$

With R being an upper bound of the kernel used and $\alpha_i^0, f^0(x_i)$ are the solution of the optimization problem described in (1) and the decision function respectively. In our case, the kernel described in (3) has an upper bound of 1. So, there is an upper bound of the leave-one-out error that can be calculated as:

$$\widehat{LOO} = \frac{1}{l} \sum_i u\left(2\alpha_i^0 - y_i f^0(x_i)\right) \ i \in V \ . \tag{5}$$

where $u(.)$ is the step function.

3 The Proposed ACO Procedure

For a given dataset $\boldsymbol{x_i} \in \mathrm{R}^n$, $i = 1, ..., l$ we have to find the best combination of $\boldsymbol{\gamma} \in \mathrm{R}^n$, $\gamma_i \in [0, 1]$ that minimizes the leave-one-out error. The proposed ACO procedure has the following elements:

- A number of artificial ants. Each ant travels across a path associated to a solution of the $\boldsymbol{\gamma}$ vector.
- For each feature i, there are m possible states of γ_i. The continuous space [0,1] is divided into m discrete values. S_j^i is the j-th state associated to the feature i. There is an initial state S^0 where all the ants start the travel.
- From one state j to another state z, there is a path that contains the following elements:
  - The propability p_{jz} of that path to be chosen by an ant.
  - A pheromone value τ_{jz} that depends on the amount of ants that have travel across this path.

Every ant begins its travel at the initial state S^0 where it can choose between m possible paths that will arrive to a state S_j^1. With this initial movement the ant has selected the value of γ_1. The choice is done with a probability p_{0j}. Once an ant has reached a state S_j^i it can only move to a state S_z^{i+1} and it will take that path with a probability p_{jz}. When the ant is in a state S^n it has reached the end of the travel and it has a possible solution of the γ vector. At every instant of time t, there are k concurrent ants that are traveling across the states. In Fig.(1) it is shown a problem with 4 features, where the scaling factor values have 5 steps. Then, an ant starting in the initial node can choose in every state between the paths drawn in dashed line, and in this example it has selected the

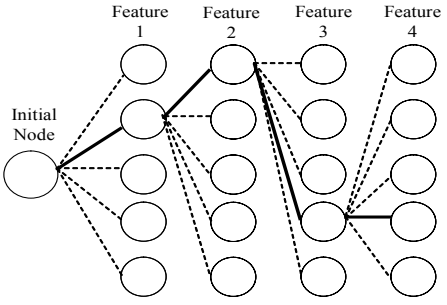

Fig. 1. An example of the possible paths an ant can choose in a 4-feature problem

continuous line. Once all the ants have finished their routes, the estimator of the leave-one-out described in (5) is evaluated for each ant.

It is important to note that we have proposed in this work to take into account only discrete values of the continuous space. The main reason for doing this, is that similar values of $gamma_i$ should give similar results.

The basic algorithm is summarized in the following steps:

1. Create the available states. In the first iteration set a uniform distribution for all probabilities.
2. Set the same amount of pheromone τ_{jz} in all the possible paths.
3. Set k ants at the initial node. Each ant will select a path to the state j with a probability p_{0j}. Once the ant is in the state j the choice of the next path is done depending on the probability of each path.
4. Once the k ants have finished their travel, compute for each of them the estimator of the leave-one-out $\widehat{LOO}_k$, as is defined in (5). Select the B ants that have achieved the lowest values in this iteration. If there are two ants with the same evaluation function value, it is first selected the one with less number of support vectors.
5. Increment the pheromone value of the tail following:

$$\Delta\tau_{jz} = \begin{cases} \frac{Q}{\widehat{LOO}_k} \; if \; k \in B \\ 0 \; if \; k \notin B \end{cases} \quad . \tag{6}$$

   where Q is a constant that has to be adjusted depending on the problem under study.
6. If the winning path until this moment has not be included in the previous point, include it now.
7. Recalculate all the path probabilities:

$$p_{jz} = \frac{\tau_{jz}^{\lambda}}{\sum \tau_{j}^{\lambda}} \quad . \tag{7}$$

   Where λ is a constant to be adjusted.

8. Evaporate pheromone values:

$$\tau_{jz}^{it+1} = (1-\rho)\,\tau_{jz}^{it} \quad . \tag{8}$$

   Where ρ, is the evaporation coefficient.
9. If the number of iterations has reached the maximum number of iterations allowed, then finish and return the path value of the ant with lowest leave-one-out value. If the number of iterations has not reached the maximum, repeat from step 3.

4 Results and Discussion

4.1 Standard Datasets

Before testing the proposed method on electronic nose datasets, we have tested it on some datasets obtained from the UCI repository database. The main reason for doing this is the high number of features the electronic nose datasets have. In order to have some control on the convergence rate, number of ants per population and some other parameters like the described constants (Q, λ, ρ) it is very useful to test on some datasets with less features in such a way that it is possible to calculate the absolute minima of the problem.

In Fig.(2) it is shown the mean of the standard deviation of the paths selected by the ants against the number of generations.It is clear that after a number of iterations, all the ants in the generation follow the same path. If this number of iterations is very low, there is a high risk of falling into a local minima. However, if the standard deviation of the solution keeps high after a number of generations, it would mean that the algorithm is not searching towards the solution, but it searches in a randomize way. In this figure it is shown different curves for different values of Q and it can be appreciated the importance of choosing an adequate value of Q. To make this figure, we have executed the algorithm one hundred of times per each value of Q, drawing the mean of the standard deviations calculated. Repeating this process for several datasets, we found that a good value of Q can be obtained calculating the mean of the the evaluation function in the first generation and dividing it by two.

One important measure for our experiments is the number of generations needed to reach the absolute minima and how many times is it reached. A measure related to this is the indicator M, that is defined as the number of generations needed to have at least a success in 80% of the tests. This indicator measures how fast the algorithm converges to the optimal solution. In Fig.(3) is represented the percentage of success against the number of generations used for the Indian-Diabetes dataset with 60 ants per generation, in continuous line, and 15 ants per generation in dashed-dot line. For this test the constants have the following values: $\rho = 0.8$, $\lambda = 2$, and $Q = 100$. It can be appreciated that $M = 45$ in the case of 60 ants per generation but in the case of 15 ants per generation this value is not obtained, since many times the algorithm falls into a local minima and the maximum success obtained is a 63%.

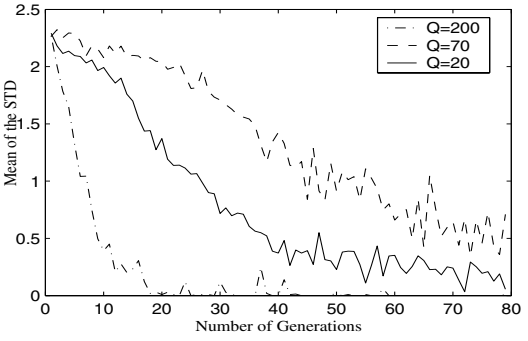

Fig. 2. Standard Deviation against number of generations for different values of Q

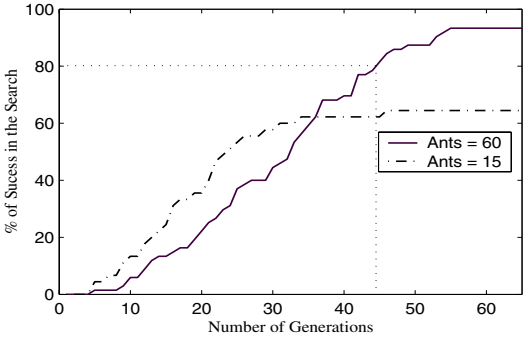

Fig. 3. Indian-Diabetes Dataset. Success for two different size populations.

In Table(1) are shown the values for several standard datasets. The Q parameter was always adjusted as explained, with $\rho = 0.9$ and $\lambda = 2$. In all cases, the exact minima was obtained through exhaustive search since although the number of combinations is big it is possible to obtain it, due to the reduced number of features.

It can be appreciated how the algorithm has a good behavior in the first and the third datasets, whereas in the second case the percentage of success is so low that the M parameter could not be measured although the test was done increasing a lot the number of generations. This can be explained because the leave-one-out estimator gives for this dataset almost plain values and the search is not optimized. These measures are interesting to work with the algorithm in a real system.

4.2 Electronic Nose Datasets

Once we have tested the method with standard datasets we can focus on the electronic nose datasets. In this case we have tested 3 datasets that were obtained

Table 1. M value for different datasets

DATASET	N Features	N of γ steps	Ants per Generation	Max. Iter Allowed	% Sucess	M Parameter
Indian-Diabetes	8	4	30	200	89	60
			30	50	75	NA
			60	200	94	45
			60	50	85	45
Breast Cancer	9	4	30	200	43	NA
			30	50	35	NA
			60	200	55	NA
			60	50	35	NA
Ecoli	5	5	30	200	90	29
			30	50	83	29
			60	200	94	27
			60	50	86	27

Table 2. M value for Electronic Nose datasets

DATASET	N Features	N of γ steps	Ants per Generation	Max. Iter Allowed	% Sucess	M Parameter
RonVsWhisky	88	10	30	200	95	31
			30	50	89	31
			60	200	97	28
			60	50	92	28
EthanolVsAlcohols	88	10	30	200	83	94
			30	50	67	NA
			60	200	85	78
			60	50	71	NA
$COVsNO_2$	88	10	30	200	78	NA
			30	50	58	NA
			60	200	84	150
			60	50	65	NA

in our laboratory using SnO_2 sensors with thermomodulation [4]. The first one is composed on some samples from a whisky and some others from Ron. The second one, is composed from composed from Ethanol and some other alcohols like methanol and propanol. The third one takes samples from CO and NO_2. Table (2) shows the results obtained for these datasets.

One of the relevant issues from the information obtained with these measures is that the algorithm works better for a population of 60 ants rather than for 30 ants. The main reason is that ants works under a cooperation way and, especially at the beginning there are more solutions explored. However, we can not conclude that the higher the number of ants the faster the convergence is.

5 Conclusion

In this work we have adapted the ACO method to optimize modified RBF kernels. This procedure is extremely important to know what features should be enhanced in electronic nose applications. The proposed method reach to a global minima with a high level of success in most of the datasets tested if the necessary parameters are well adjusted. Future work will explore possible attractiveness functions, new datasets and modifications of the ACO algorithm to work with continuous spaces.

Acknowledgement. This work was supported by Comunidad of Madrid project CAM-UAH 2005/031.

References

1. Hines, E., Llobet, E., Gardner, J.: Electronic noses: a review of signal processing techniques. IEE Proc. Circuits Dev.and Systems **146** (1999) 297–310
2. Distante, C., Leo, M., Siciliano, P., Persaud, K.: On the study of feature extraction methods for an electronic nose. Sensors and Actuators B: Chem. **87** (2002) 274–288
3. Vapnik, N.V.: The Natureof Statistical Learning Theory. Springer-Verlag, New York. (2000) 1ed: 1998.
4. Al-Khalifa, S., Maldonado, S., Gardner, J.: Identification of co and no2 using a thermally resistive microsensor and support vector machine. IEE Proc. Science Meas. and Tech. **150**(6) (2003) 11–14
5. Pardo, M., Sberveglieri, G., Gardini, S., Dalcanale, E.: Classification of electronic nose data with support vector machines. Sensors and Actuators B: Chem. **107** (2005) 730–737
6. Chapelle, O., Vapnik, V., Bousquet, O., Mukherjee, S.: Choosing multiple parameters for support vector machines. Machine Learning **46**(1) (2002) 131–159
7. Dorigo, M., Stützle, T.: Ant Colony Optimization. MIT Press (2004)
8. Dorigo, M., Di Caro, G.: The ant colony optimization meta-heuristic. In Corne, D., Dorigo, M., Glover, F., eds.: New Ideas in Optimization. McGraw-Hill, London (1999) 11–32
9. Platt, J.: Fast training of svms using sequential minimal optimization. In Schölkopf, B., Burges, C., Smola, A., eds.: Advances in Kernel Methods – Support Vector Learning. MIT Press (1998) 185–208
10. Joachims, T.: Estimating the generalization performance of a SVM efficiently. In Langley, P., ed.: Proc. of ICML-00, Morgan Kaufmann , San Francisco, US (2000) 431–438