DOI: 10.1007/s12541-012-0096-1

Feasibility Classification of New Design Points Using Support Vector Machine Trained by Reduced Dataset

Seung-Hyun Jeong[1], Dong-Hoon Choi[2] and Minjoong Jeong[3,#]
1 Graduate School of Mechanical Engineering, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, South Korea, 133-791
2 The Center of Innovative Design Optimization Technology, Hanyang University, 222, Wangsimni-ro, Seongdong-gu, Seoul, South Korea, 133-791
3 Advanced Research Team, Korea Institute of Science and Technology Information, 335, Gwahangno, Yuseong-gu, Daejeon, South Korea, 305-806
Corresponding Author / E-mail: jeong@kisti.re.kr, TEL: +82-42-869-0632, FAX: +82-42-869-0599

KEYWORDS: Support vector machine (SVM), Feasibility classification, K-means clustering, Air-conditioner pipe design problem

In this paper, we propose to use a support vector machine (SVM) for the classification of design data. Although the SVM is a very popular technique in data mining, it is rarely applied to an industrial design process that may require information regarding the feasibility of the design point of interest. To check the feasibility, the designer must conduct experiments or computer simulations, which may incur considerable cost. Therefore, the SVM can be an effective tool for predicting feasible and infeasible regions because it only uses the cumulative design data. In this paper, we used the SVM to classify sample datasets drawn from mathematical test problems and from an air-conditioner pipe design example. Our results indicate that the SVM is capable of very accurately identifying feasible and infeasible regions in the design space. Further, we were able to reduce the training time of the SVM by using the k-means clustering algorithm to reduce the amount of training data, taking advantage of the powerful generalization abilities of the SVM. Consequently, we conclude that the SVM can be an effective tool to assess feasibility at certain design points, avoiding some of the high computational costs of the analysis.

Manuscript received: September 6, 2011 / Accepted: December 25, 2011

1. Introduction

In this paper, we propose the use of a support vector machine (SVM) to check whether a design point of interest can satisfy or not satisfy the design requirements, in other words, the feasibility of the design point of interest. If the designer desires to know the feasibility of a design point of interest, it is common to conduct experiments or simulations that may require considerable cost or simulation time. However, if we obtain the cumulative design data beforehand, it is wasteful to leave the data without reusing in the design process. Therefore, a method to reuse the cumulative design data obtained beforehand needs to be developed. Although some researchers propose to use SVM for design of electric device[1] and chemical compound,[2] it is difficult to find practical SVM applications on the field of engineering design problem.

The SVM is a classification method introduced in 1992 by Boser, Guyon, and Vapnik.[3] Because of its high accuracy with high-dimensional data, the SVM is used widely in bioinformatics,[4] text classification,[5] face detection in digital images,[6] and other applications. Also, the SVM can be an appropriate technique for the parameter identification[7-9] which is the one of critical issues of precision engineering and manufacturing process of mechanical systems.

The training algorithm of the SVM detects optimal separating hyper-planes by maximizing the margin between two datasets. Meanwhile, decision boundaries in the design space must exist because the feasible regions in the design space can be separated by the limit value of the constraint function. Because of the basic concept of finding the separating hyper-planes, the SVM may be an appropriate tool to divide a design space into feasible and infeasible regions.

Although the SVM has been shown to be highly accurate, training algorithm involves the solution of constrained quadratic programming (QP) problems and therefore requires a large amount of memory and significant training time for training large datasets. In contrast, the patterns that are actually used for producing final decision functions are a small subset of the training data called the support vectors. Therefore, to reduce the training time, a number of methods have been proposed to identify appropriate support vectors among training data, including methods using the neighborhood property[10] and the *k*-means clustering algorithm.[11-13] The data from the design process tends to be distributed densely in certain regions;

hence, we applied a k-means clustering algorithm because it is more suitable for dense training datasets.[11] This allows us to use subsets of the training data to calculate the final feasibility decision functions, thus significantly reducing the training time while maintaining the accuracy of the SVM.

To test the accuracy of the proposed SVM, we applied it to sample data obtained from several mathematical problems[14] and a practical air-conditioner pipe design example.[15] The data from each problem are divided into two groups for training and predicting, and used for evaluating the accuracy of the proposed SVM.

The rest of this paper is organized as follows: In Section 2, the basic SVM theory is briefly described. In Section 3, we describe the proposed SVM that uses the k-means clustering algorithm to select the training data. The eight mathematical problems that we used for testing the SVM and their feasibility classification results are presented and explained in Section 4. The practical air-conditioner pipe design example and the related test results are presented and explained in Section 5. Finally, the concluding remarks are given in Section 6.

2. Formulation of support vector machine

In this section, we introduce the basic SVM formulation for a linear two-class classifier and then extend it to a non-linear classifier through the use of a kernel function.

2.1 Linear classifiers

Given a training set of instance-label pairs ($\mathbf{x}_i$, y_i), $i = 1, \ldots, n$, where $\mathbf{x}_i \in R^n$, and $y_i \in \{1,-1\}$, the linear classifier is based on a linear discriminant function of the form:

$$f(\mathbf{x}) = \mathbf{w}^T\mathbf{x} + b \tag{1}$$

The vector $\mathbf{w}$ is known as the weight vector, and b is called the bias. The sign of this function determines the two groups into which the data are classified.

First, we will begin with the simplest case of linear classifiers trained on separable data. This is presented in Fig. 1. Suppose that all the training data satisfy the following constraints:

$$\mathbf{x}_i\mathbf{w} + b \geq +1 \quad \text{for} \quad y_i = +1 \tag{2}$$

$$\mathbf{x}_i\mathbf{w} + b \leq -1 \quad \text{for} \quad y_i = -1 \tag{3}$$

These can be replaced with one inequality:

$$y_i(\mathbf{x}_i\mathbf{w} + b) - 1 \geq 0 \tag{4}$$

The perpendicular distance from the hyperplane H_1: $\mathbf{x}_i\mathbf{w} + b = 1$ to the origin is $|1 - b|/||\mathbf{w}||$. Similarly, the perpendicular distance from the hyperplane H_2: $\mathbf{x}_i\mathbf{w} + b = -1$ to the origin is $|-1 - b|/||\mathbf{w}||$. Note that H_1 and H_2 are parallel, and the distance between them is $2/||\mathbf{w}||$. The training algorithm simply looks for the separating hyperplane with the largest margin by maximizing the distance $2/||\mathbf{w}||$. This leads to the following constrained optimization problem:

$$\text{Minimize} \quad ||\mathbf{w}||^2/2 \tag{5}$$

$$\text{Subject to} \quad y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1, \ i = 1, \ldots, n \tag{6}$$

Secondly, we will consider the non-separable case, which is presented in Fig. 2. In this case, we introduce the positive slack variables ($\xi_i \geq 0$), $i = 1, \ldots, n$ in the constraints, which will become

$$\mathbf{x}_i\mathbf{w} + b \geq +1 - \xi_i \quad \text{for} \quad y_i = +1 \tag{7}$$

$$\mathbf{x}_i\mathbf{w} + b \leq -1 - \xi_i \quad \text{for} \quad y_i = -1 \tag{8}$$

In this case, the optimization problem now becomes

$$\text{Minimize } ||\mathbf{w}||^2/2 + C\sum_{i=1}^{n}\xi_i \tag{9}$$

$$\text{Subject to } y_i(\mathbf{w}^T\mathbf{x}_i + b) \geq 1 - \xi_i \, \xi_i \geq 0 \qquad i = 1, \ldots, n \tag{10}$$

This formulation is called the soft-margin SVM.[16] According to the Lagrange multipliers method, the dual formulation can be expressed in terms of the variable α_i as follows:[4,16,17]

$$\text{Maximize} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n}y_i y_j \alpha_i \alpha_j \mathbf{x}_i^T\mathbf{x}_j \tag{11}$$

$$\text{Subject to} \quad \sum_{i=1}^{n}y_i\alpha_i = 0 \qquad 0 \leq \alpha_i \leq C \tag{12}$$

After solving this dual formulation, we can express the weight vector in terms of the input patterns as follows:

$$\mathbf{W} = \sum_{i=1}^{n}y_i\alpha_i\mathbf{x}_i \tag{13}$$

The patterns $\mathbf{x}_i$ for which $\alpha_i > 0$ are called support vectors.

2.2 Non-linear classifiers

To generalize the non-linear case, we introduce the kernel function that replaces the dot product in linear classifiers. Therefore,

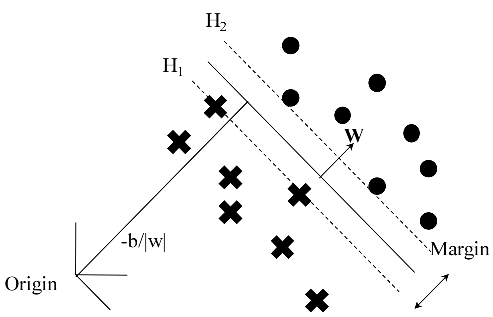

Fig. 1 Linear classifiers trained on separable data

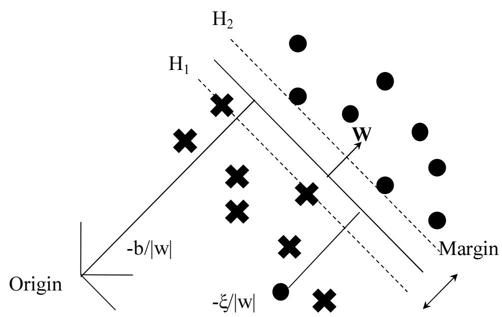

Fig. 2 Linear classifiers trained on non-separable data

the dual formulation will be changed as follows:[4,16,17]

$$\text{Maximize} \quad \sum_{i=1}^{n}\alpha_i - \frac{1}{2}\sum_{i=1}^{n}\sum_{j=1}^{n} y_i y_j \alpha_i \alpha_j K\left(\mathbf{x}_i, \mathbf{x}_j\right) \tag{14}$$

$$\text{Subject to} \quad \sum_{i=1}^{n} y_i \alpha_i = 0 \quad 0 \le \alpha_i \le C \tag{15}$$

In general, two functions are used as the kernel functions:

$$\text{Polynomial: } K(\mathbf{x}_i,\mathbf{x}_j) = (\gamma \mathbf{x}_i \mathbf{x}_j + 1)^d \tag{16}$$

Radial Basis Function (RBF):

$$K(\mathbf{x}_i,\mathbf{x}_j) = \exp(-\|\mathbf{x}_i - \mathbf{x}_j\|^d / 2\sigma^2) \tag{17}$$

In the RBF kernel, if $d = 1$, it is the exponential RBF; otherwise, if $d = 2$, it is the Gaussian RBF. The degree of the polynomial kernel d and the width parameter of the RBF kernel σ have a significant effect on the flexibility of the resulting classifiers. As the degree of the polynomial kernel is increased, the curvature of the decision boundary is also increased. Further, when the width parameter of the RBF kernel σ is decreased, the locality of the SVM increases, leading to a greater curvature of the decision boundary.[4]

3. Support vector machine with k-means clustering

Training of the SVM is accomplished through constrained quadratic programming (QP). Solving the QP problem requires significant memory and training time if the number of patterns is large. Compared with the time required for the analysis at a design point, the training time is short and hence can be ignored. However, in the design process, more data can be generated; further, finding a kernel function with an appropriate kernel parameter requires several training processes. Therefore, if the number of training patterns is decreased, it will be helpful to find an appropriate kernel and its parameter in a short time.

The training patterns for the SVM can be divided into three data sets according to the Lagrange multiplier values. The first set has ordinary vectors, where $\alpha_i = 0$. The second set has support vectors on the margin, $0 < \alpha_i < C$. The last set has the support vectors between the margin, $\alpha_i = C$. Therefore, we can guess how the patterns near the decision boundary affect the decision function. Thus, we select these separable data a priori using the *k*-means clustering algorithm.

3.1 K-means clustering

K-means clustering is a partitional clustering algorithm[18] and is accomplished using the procedure described below. An illustrative procedure is also described in Fig. 3.

Step 1: Select an initial partition with *K* clusters.

Step 2: Generate a new partition by assigning each pattern to its closest cluster center.

Step 3: Compute new centers as the centroids of the clusters.

Step 4: Repeat steps 2 and 3 until no cluster centers change positions during step 3.

As the selection of initial centers significantly affects the final clusters, we repeat this procedure 100 times with different initial centers for each trial. A good cluster set minimizes the sum of distances from each cluster's data to its cluster center and maximizes the sum of the distances between the cluster centers. Therefore, we select a cluster set that has the minimum value mentioned (18) in 100 repetitions. Further, in this paper, we use *k*-means clustering to reduce the training dataset, not to find the cluster; hence, the appropriate number of clusters is not studied seriously. Therefore, we fixed the number of clusters as 100.

$$F = \sum_{i=1}^{NC}\sum_{j=1}^{nc_i} d\left(\mathbf{x}_{ij}, \mathbf{m}_i\right) / \sum_{i=1}^{NC}\sum_{j>i}^{NC} d\left(\mathbf{m}_i, \mathbf{m}_j\right) \tag{18}$$

where NC = number of clusters

nc_i = number of patterns in i^{th} cluster

$\boldsymbol{m}_i$ = i^{th} cluster center

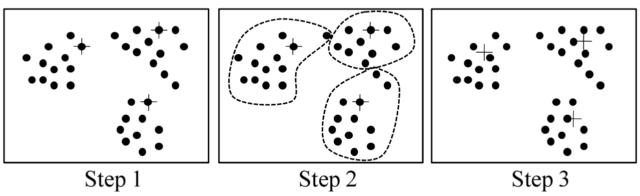

Fig. 3 Illustrative procedure of *k*-means clustering

3.2 Training data reduction with k-means clustering

We select the training data from the clustering result. As shown in Fig. 4, if the dataset in a cluster has two labels, we can guess that this dataset is near the decision boundary. In this case, we select all of the data in the cluster as a training set. On the other hand, in a cluster with a dataset with only one label, we select one datum that has the shortest distance from its center for training. If we select only the data close to the decision boundary, the accuracy of the SVM can be deteriorated in the entire design space. Therefore, we select one datum in the cluster with one label. The entire procedure is mentioned below:

Step 1: Make a cluster set using the procedure mentioned in 3.1.

Step 2: Reduce the training data according to the label at each cluster.

Step 3: Use the reduced data as a training set in the SVM.

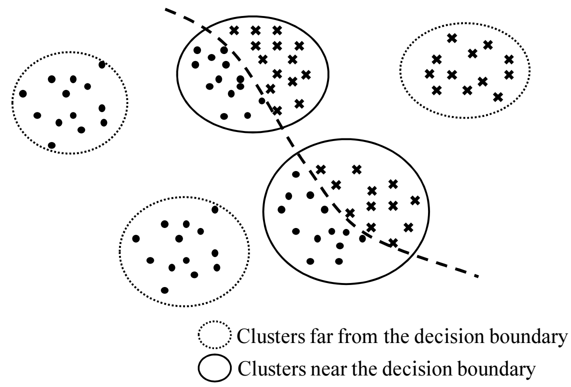

Fig. 4 Selection of training dataset near decision boundary through *k*-means clustering

4. Application of SVM to eight mathematical problems

To verify the accuracy of the SVM, we used data drawn from eight mathematical problems.[14] The data from each problem are distributed randomly in the design space and classified as feasible or not according to the satisfaction of the constraint functions. The data are divided into two groups for training and prediction and scaled in the range of [0, 10]. To solve the QP in the training algorithm, we used the MATLAB Quadratic Programming solver.[19]

4.1 Data

We obtained artificial data distributed randomly in the design space, and the composition of the resulting dataset is shown in Table 1. The training dataset was significantly reduced after the selection of training data using the *k*-means clustering algorithm. This reduced set is given in Table 1.

4.2 Results

We evaluated the accuracy of the SVM for seven kernel functions. The kernels included three types of polynomial kernels and four types of RBF kernels. Each polynomial kernel has a different value of *d*, and each RBF kernel has a different value of σ. We measured the accuracy of the SVM by the percentage of correctly classified data included in the predicted data. We evaluated the accuracy of the SVM using all the training data and the reduced training data. Tables 2 and 3 illustrate the accuracy of the SVM trained by a complete set of training data, and Tables 4 and 5 illustrate the accuracy of the SVM trained using a reduced set of training data. For the mathematical optimization problems, we predicted 5000 random data in each problem's design space because it did not require a high computational cost.

In Tables 2 and 3, we present the accuracy of the SVM for both training and prediction. Because each problem incorporated some nonlinearity, the accuracy improved when the parameter of the polynomial kernel was increased from $d = 1$ to 2. However, when we increased the parameter d so that it was larger than 2, the accuracy improved only slightly or not at all. Moreover, for the RBF kernel, the accuracy increased as the value of parameter decreased, and the accuracy was best at $\sigma = 0.1$. However, at $\sigma = 0.025$, the prediction accuracy was very low although the training accuracy was very high, as the kernel function's value was almost zero, causing the SVM to find trivial solutions that overfit the training data.

In Tables 4 and 5, a tendency of the kernel function similar to

Table 1 Dataset from eight mathematical problems

	ndv	ncon	Size of dataset (feasible/ infeasible)	Training dataset (feasible/ infeasible)	Reduced training dataset (feasible/ infeasible)
(1)	4	3	1082 (386/696)	800 (280/520)	247 (151/96)
(2)	4	6	1073 (453/620)	800 (329/471)	220 (122/98)
(3)	7	4	1036 (24/1012)	800 (10/790)	132 (10/122)
(4)	8	4	1290 (186/1104)	1000 (125/875)	398 (125/273)
(5)	9	13	1400 (277/1123)	1000 (168/832)	364 (168/196)
(6)	10	8	1253 (131/1122)	1000 (100/900)	336 (96/240)
(7)	10	8	1211 (192/1019)	1000 (144/856)	428 (144/284)
(8)	13	13	1271 (64/1207)	1000 (63/937)	273 (63/208)

ndv: number of design variables, ncon: number of constraints

Table 2 Accuracy of SVM trained by all training data, using polynomial kernel

		$d = 1$	$d = 2$	$d = 3$	$d = 4$
(1)	Training	58.50	100.00	100.00	100.00
	Prediction	55.32	96.81	96.45	96.45
(2)	Training	97.62	99.50	99.75	100.00
	Prediction	94.51	96.34	97.80	97.07
(3)	Training	98.00	98.75	98.75	98.75
	Prediction	91.10	92.80	94.49	94.49
(4)	Training	88.20	98.30	98.80	98.70
	Prediction	78.97	97.59	95.52	96.21
(5)	Training	86.40	100.00	100.00	100.00
	Prediction	79.25	96.25	96.50	95.75
(6)	Training	72.30	99.80	100.00	100.00
	Prediction	35.18	95.65	94.47	94.47
(7)	Training	85.00	99.20	99.80	94.90
	Prediction	74.88	93.36	90.52	83.41
(8)	Training	92.90	93.60	93.60	98.90
	Prediction	99.63	98.89	99.63	99.63

Table 3 Accuracy of SVM trained by all training data, using RBF kernel

		$\sigma = 0.025$	$\sigma = 0.1$	$\sigma = 0.25$	$\sigma = 0.5$	$\sigma = 1.0$
(1)	Training	100.00	100.00	100.00	100.00	47.50
	Prediction	69.86	95.74	96.45	90.07	50.35
(2)	Training	100.00	100.00	100.00	100.00	100.00
	Prediction	70.70	97.07	96.34	95.97	95.97
(3)	Training	100.00	100.00	65.63	65.75	65.75
	Prediction	82.63	86.52	82.63	77.12	42.37
(4)	Training	100.00	100.00	100.00	100.00	73.00
	Prediction	39.66	96.21	93.45	88.97	73.79
(5)	Training	100.00	100.00	100.00	100.00	21.00
	Prediction	37.25	95.75	91.75	79.75	35.00
(6)	Training	100.00	100.00	100.00	100.00	100.00
	Prediction	68.77	96.44	95.26	91.30	78.26
(7)	Training	100.00	100.00	100.00	100.00	100.00
	Prediction	33.18	93.36	88.15	78.67	72.04
(8)	Training	100.00	100.00	100.00	100.00	44.80
	Prediction	27.68	99.26	98.52	96.31	42.44

Table 4 Accuracy of SVM trained by reduced data, using polynomial kernel

		$d = 1$	$d = 2$	$d = 3$	$d = 4$
(1)	Training	35.63	100.00	100.00	100.00
	Prediction	51.42	93.97	94.33	95.39
(2)	Training	86.36	97.73	99.07	100.00
	Prediction	93.77	94.51	95.97	96.70
(3)	Training	90.15	91.67	91.67	91.67
	Prediction	91.53	93.64	93.64	93.64
(4)	Training	68.34	95.73	96.73	96.73
	Prediction	78.28	97.24	94.83	95.86
(5)	Training	78.02	100.00	100.00	100.00
	Prediction	76.50	96.25	96.50	95.75
(6)	Training	86.01	100.00	100.00	100.00
	Prediction	66.01	86.56	87.35	90.51
(7)	Training	88.55	98.36	99.53	99.53
	Prediction	84.83	88.63	84.36	82.94
(8)	Training	74.54	76.38	76.38	95.94
	Prediction	99.63	98.89	99.63	99.63

that seen in Tables 2 and 3 is illustrated. Compared to the accuracy of the method that uses all of the training data, the accuracy was slightly lower but the amount of training data required was dramatically reduced. We were still able to evaluate the feasibility of design points with reduced training data.

Table 6, which describes the accuracy of the SVM when we predicted 5000 random data in the design space for each mathematical optimization problem, provides more evidence of the generalization capabilities of the proposed SVM.

As illustrated in Table 6, the accuracy was very high except for the RBF kernel at σ = 1.0. Accuracy is higher than 95% in most cases, especially for the polynomial kernel with d = 2 and the RBF kernel with σ = 0.1 or 0.25. Therefore, the proposed SVM proved very effective for the feasibility classification.

5. Application of SVM to air-conditioner pipe design problem

The air-conditioner pipe design problem[15] that we used for testing the proposed SVM incorporates five design variables representing the lengths of two pipes (Fig. 5). The constraints were three quality-related performance measures: natural frequency, maximum stress, and vibration fatigue life. We have to design the natural frequency of the pipe to avoid the first, second, and third rotating frequencies of the compressor. To address this constraint, we applied a penalty value if the natural frequency of the pipe was in the range of the avoided frequency. Further, we applied the weight value for the first, second, and third ranges of the avoiding frequency. Therefore, the final penalty value is the sum of the products of the penalties and the weight values as shown in (19). We considered the maximum stresses in three dimensions, so there were five constraints in total. In this problem, the pipe mass depended linearly on the design variables, and the designers had to be careful to satisfy design constraints. Therefore, this example was a good test of the SVM.

An automated multi-disciplinary analysis system was built in PIAnO[20] and is shown in Fig. 6. First, using a morphing technique, the analysis system automatically generated a finite element model according to the specified design variables. The function values of the performances were then evaluated through an analysis of the finite element model with appropriate computer-aided engineering (CAE) software. The entire process was completed automatically, and the analysis system became a black box with inputs and outputs.

Air conditioner pipe design problem: (19)

Penalty value,	$P \le 0.5$
Maximum stress (x)	$\le$ 190 MPa
Maximum stress (y)	$\le$ 190 MPa
Maximum stress (z)	$\le$ 190 MPa
Fatigue Life	$\ge$ 1.5 hours

$-56 \le x_1 \le 50$,
$-100 \le x_2 \le 50$,
$-20 \le x_3 \le 10$,
$-60 \le x_4 \le 15$,
$-22 \le x_5 \le 20$

Table 5 Accuracy of SVM trained by reduced data, using RBF kernel

		σ = 0.025	σ = 0.1	σ = 0.25	σ = 0.5	σ = 1.0
(1)	Training	100.00	100.00	100.00	100.00	65.99
	Prediction	51.77	96.45	95.04	83.69	50.35
(2)	Training	100.00	100.00	100.00	100.00	100.00
	Prediction	57.14	96.70	95.97	94.14	91.58
(3)	Training	100.00	100.00	46.97	47.73	45.45
	Prediction	85.17	84.75	79.24	73.73	46.19
(4)	Training	100.00	100.00	100.00	100.00	53.77
	Prediction	32.07	96.21	94.14	88.28	72.07
(5)	Training	100.00	100.00	100.00	100.00	49.73
	Prediction	34.00	92.75	90.50	77.00	34.75
(6)	Training	100.00	100.00	100.00	100.00	100.00
	Prediction	27.67	92.89	91.30	45.45	27.27
(7)	Training	100.00	100.00	100.00	100.00	100.00
	Prediction	30.33	90.05	87.20	74.88	64.93
(8)	Training	100.00	100.00	100.00	100.00	58.30
	Prediction	2.21	98.89	97.42	93.36	42.80

Table 6 Accuracy of SVM trained by reduced data and predicting 5000 random data

	Polynomial			RBF			
	d = 1	d = 2	d = 3	σ = 0.1	σ = 0.25	σ = 0.5	σ = 1.0
(1)	85.16	99.42	99.60	98.72	98.16	92.64	41.28
(2)	99.62	94.90	90.66	99.64	99.54	99.10	98.86
(3)	97.90	97.02	97.30	100.00	73.00	75.56	83.20
(4)	99.84	97.86	99.14	99.90	99.90	99.54	96.82
(5)	96.34	99.56	99.66	98.46	97.50	86.08	2.74
(6)	99.54	85.96	86.36	99.12	98.62	96.48	81.48
(7)	99.28	98.68	97.70	98.16	97.44	92.06	68.54
(8)	99.98	100.00	99.98	100.00	100.00	99.36	85.00

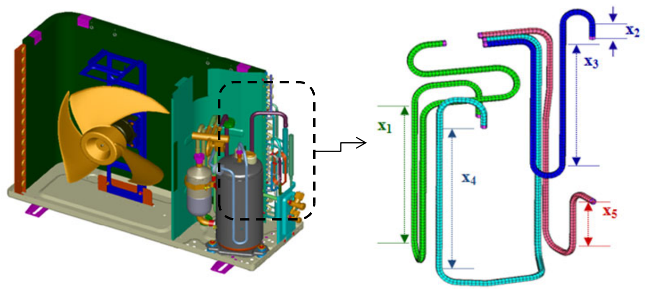

Fig. 5 Analysis model and five design variables of air-conditioner pipe design problem

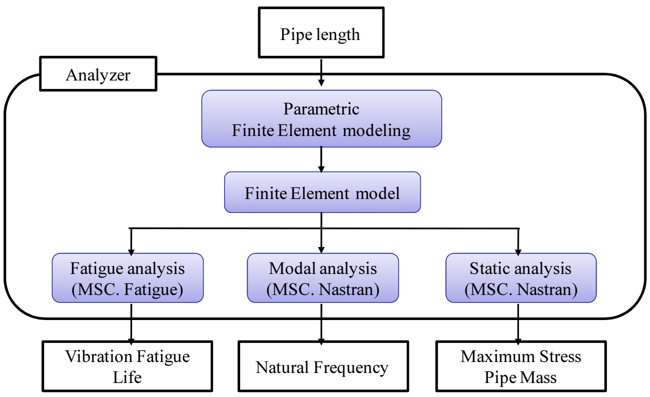

Fig. 6 Integration of multi-disciplinary design analysis

where $P = \sum_{i=1}^{N} \omega_i p_i$,

P = Total penalty value,

p_i = Penalty value of each frequency,

ω_i = weight value, i = 1, 2, 3

5.1 Data

The composition of data obtained in the design process of the air-conditioner pipe is shown in Table 7. The data are divided for training and prediction randomly, and the training data are significantly reduced through the proposed training data reduction method using the *k*-means clustering algorithm.

5.2 Results

We present three cases in Table 8. The first case is an evaluation of the accuracy of the SVM trained by all the training data, and the second and the third cases represent the accuracy of the SVM when trained using a reduced dataset. The second and the third cases differ in the composition of the dataset used for prediction. In the third case, we added the data remained after the selection of the training dataset to the prediction dataset. Therefore, the remaining dataset which may not have been close to the decision boundary was added to the dataset for prediction in the case. Except in the case of the polynomial kernel with d = 1 and 3, the accuracy of prediction in the third case was higher than in the second case. Therefore, if we were to select a design point adjacent to the decision boundary, this procedure may not yield good classification results. However, except for these special cases, we can obtain the feasibility information with high accuracy.

6. Conclusions

We proposed the use of a SVM to predict the feasibility of new design points without computationally expensive function evaluations. We found that the proposed SVM requires a significant amount of training time as the amount of training data increases. We reduced the training datasets by identifying and selecting data that are most likely to be near the decision boundary. Using the *k*-means clustering algorithm, we could reduce the training datasets in a manner that allowed us to guarantee the generalization capabilities of the SVM. Through several mathematical problems and air-conditioner pipe design problem, we identified the tendency of accuracy according to the kernel types and their parameters. The RBF kernel yields a relatively high accuracy when the value of the parameter is decreased. However, if the parameter has a very low value, the value of the kernel function approaches zero, with the result that the SVM produces trivial solutions that overfit the training data, and its generalization capability deteriorates because of the overfitting.

To evaluate the accuracy of the proposed SVM, we applied it to data from several mathematical optimization problems and a practical air-conditioner pipe design problem. The results of these experiments demonstrate the SVM's ability to predict the feasibility of the new design points. Therefore, the proposed SVM is applicable to practical design problems. Using the SVM, we could build discriminant functions and predict feasibility without expensive function evaluations of new design points. Moreover, using the *k*-means clustering algorithm, it is possible to reduce the size of the training datasets while maintaining the generalization capabilities of the SVM.

Table 7 Dataset from air-conditioner pipe design problem

Size of dataset (feasible/infeasible)	Training dataset (feasible/infeasible)	Reduced training dataset (feasible/infeasible)
1300 (577/723)	1000 (464/536)	481 (211/270)

Table 8 Accuracy of SVM in air-conditioner pipe design problem

		Polynomial		
		$d = 1$	$d = 2$	$d = 3$
Training set	Training (1000)	77.10	80.60	87.80
	Prediction (300)	79.33	75.30	79.00
Reduced training set	Training (481)	57.17	61.12	79.21
	Prediction (300)	60.33	73.00	68.67
	Training (481)	57.17	61.12	79.21
	Prediction (819)	51.77	84.13	64.35
		RBF		
		$\sigma = 0.1$	$\sigma = 0.25$	$\sigma = 0.5$
Training set	Training (1000)	100.00	100.00	100.00
	Prediction (300)	79.33	80.00	79.00
Reduced training set	Training (481)	100.00	100.00	100.00
	Prediction (300)	78.67	78.33	76.67
	Training (481)	100.00	100.00	100.00
	Prediction (819)	87.06	86.81	85.35

ACKNOWLEDGEMENT

This work was supported by the National Research Foundation (NRF) of Korea grant funded by the Korea government (MEST) (No. 2011-0016701). Also, this work was supported by grants from "Development of the Prototyping Ball Bearings for a Rocket Turbopump" project of Ministry of Education Science and Technology (MEST). The authors thank to MEST.

REFERENCES

1. Ding, M. and Vemur, R. I., "An Active Learning Scheme Using Support Vector Machines for Analog Circuit Feasibility Classification," Proc. of the 18th International Conference on VLSI Design, pp. 528-534, 2005.

2. Burdidge, R., Trotter, M., Buxton, B. and Holden, S., "Drug Design by Machine Learning: Support Vector Machines for Pharmaceutical Data Analysis," Computers and Chemistry, Vol. 26, No. 1, pp. 5-14, 2001.

3. Boser, B. E., Guyon, I. M. and Vapnik, V. N., "A Training Algorithm for Optimal Margin Classifiers," Proc. of the 5th Annual ACM Workshop on COLT, pp. 144-152, 1992.

4. Ben-Hur, A. and Weston, J., "A User's Guide to Support Vector Machines," Methods in Molecular Biology, Vol. 609, pp. 223-

239, 2010.

5. Joachims, T., "Transductive Inference for Text Classification using Support Vector Machines," Proc. of the 16th International Conference on Machine Learning, pp. 200-209, 1999.

6. Osuna, E., Freund, R. and Girosi, F., "Training Support Vector Machines: An Application to Face Detection," IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp. 130-136, 1997.

7. Cho, K. R., Seok, J. K. and Lee, D. H., "Applications of mechanical parameter identification with support vector machine for AC motor control system," Proc. of the 30th Annual Conference of the IEEE Industrial Electronics Society, Vol. 3, pp. 2110-2115, 2004.

8. Pan, Y. R., Shih, Y. T., Horng, R. H. and Lee, A. C., "Advanced Parameter Identification for a Linear-Motor-Driven Motion System Using Disturbance Observer," Int. J. Precis. Eng. Manuf., Vol. 10, No. 4, pp. 35-47, 2009.

9. Lee, D. M., Zhu, Z., Lee, K. I. and Yang, S. H., "Identification and Measurement of Geometric Errors for a Five-axis Machine Tool with a Tilting Head using a Double Ball-bar," Int. J. Precis. Eng. Manuf., Vol. 12, No. 2, pp. 337-343, 2011.

10. Shin, H. J. and Cho, S. Z., "Fast Pattern Selection for Support Vector Classifiers," Advances in Knowledge Discovery and Data Mining, Vol. 2637, pp. 376-387, 2003.

11. Barros de Almeida, M., De Padua Braga, A. and Braga, J. P., "SVM-KM: Speeding SVMs Learning with a Priori Cluster Selection and *k*-Means," Proc. of the 6th Brazilian Symposium on Neural Networks, pp. 162-167, 2000.

12. Koggalage, R. and Halgamuge, S., "Reducing the Number of Training Samples for Fast Support Vector Machine Classification," Neural Inform. Process. Lett. Rev., Vol. 2, No. 3, pp. 57-65, 2004.

13. Zheng, S. F., Lu, X. F., Zheng, N. N. and Xu, W. P., "Unsupervised Clustering Based Reduced Support Vector Machines," Proc. of the IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Vol. 2, pp. 821-824, 2003.

14. Hock, W. and Schittkowski, K., "Test Examples for Nonlinear Programming Codes," Springer, 1981.

15. Kim, R. H., "Design Optimization of an Air-Conditioner Piping System to Reduce the Cost of Pipes," M.Sc. Dissertation, Department of Mechanical & Industrial Engineering, Hanyang Univ., 2008.

16. Cortes, C. and Vapnik, V., "Support-Vector Networks," Machine Learning, Vol. 20, No. 3, pp. 273-297, 1995.

17. Cristianini, N. and Shawe-Taylor, J., "An Introduction to Support Vector Machines and Other Kernel-based Learning Methods," Cambridge University Press, 2000.

18. Jain, A. K. and Dubes, R. C., "Algorithms for Clustering Data," Prentice Hall College Div., pp. 89-133, 1998.

19. Mathworks Inc., "Matlab Optimization Toolbox User's Guide," 2007.

20. PIDOTECH Inc., "PIAnO (Process Integration, Automation, and Optimization) User's Manual Ver. 3.3," 2011.

APPENDIX

Eight Mathematical Problems

(1) Subject to

$$8 - x_1^2 - x_2^2 - x_3^2 - x_4^2 - x_1 + x_2 - x_3 + x_4 \geq 0$$
$$10 - x_1^2 - 2x_2^2 - x_3^2 - 2x_4^2 + x_1 + x_4 \geq 0$$
$$5 - 2x_1^2 - x_2^2 - x_3^2 - 2x_1 + x_2 + x_4 \geq 0$$

(2) Subject to

$$8 - x_1 - 2x_2 \geq 0$$
$$12 - 4x_1 - x_2 \geq 0$$
$$12 - 3x_1 - 4x_2 \geq 0$$
$$8 - 2x_3 - x_4 \geq 0$$
$$8 - x_3 - 2x_4 \geq 0$$
$$5 - x_3 - x_4 \geq 0$$

(3) Subject to

$$1 - 0.5x_1^{0.5}x_3^{-1}x_6^{-2}x_7 - 0.7x_1^3x_2x_3^{-2}x_6x_7^{0.5} - 0.2x_2^{-1}x_3x_4^{-0.5}x_6^{2/3}x_7^{1/4} \geq 0$$
$$1 - 1.3x_1^{-0.5}x_2x_3^{-1}x_5^{-1}x_6 - 0.8x_3x_4^{-1}x_5^{-1}x_6^2 - 3.1x_1^{-1}x_2^{0.5}x_4^{-2}x_5^{-1}x_6^{1/3} \geq 0$$
$$1 - 2x_1x_3^{-1.5}x_5x_6^{-1}x_7^{4/5} - 0.1x_2x_3^{-0.5}x_5x_6^{-1}x_7^{-0.5}$$
$$- x_1^{-1}x_2x_3^{0.5}x_5^{-0.65}x_2^{-2}x_3x_5x_6^{-1}x_7 \geq 0$$
$$1 - 0.2x_1^{-2}x_2x_4^{-1}x_5^{0.5}x_7^{1/3} - 0.3x_1^{0.5}x_2^2x_4^{1/3}x_7^{1/4}x_5^{-2/3}$$
$$- 0.4x_1^3x_2x_3x_5x_7^{3/4} - 0.5x_3^{-2}x_4x_7^{0.5} \geq 0$$

(4) Subject to

$$1 - 0.0588x_5x_7 - 0.1x_1 \geq 0$$
$$1 - 0.0588x_6x_8 - 0.1x_1 - 0.1x_2 \geq 0$$
$$1 - 4x_3x_5^{-1} - 2x_3^{-0.71}x_5^{-1} - 0.0588x_3^{-1.3}x_7 \geq 0$$
$$1 - 4x_4x_6^{-1} - 2x_4^{-0.71}x_6^{-1} - 0.0588x_4^{-1.3}x_8 \geq 0$$
$$0.1 \leq x_i \leq 10, \qquad i = 1, \ldots, 8$$

(5) Subject to

$$1 - x_3^2 - x_4^2 \geq 0$$
$$1 - x_9^2 \geq 0$$
$$1 - x_5^2 - x_6^2 \geq 0$$
$$1 - x_1^2 - (x_2 - x_9)^2 \geq 0$$
$$1 - (x_1 - x_5)^2 - (x_2 - x_6)^2 \geq 0$$
$$x_3x_9 \geq 0$$
$$1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0$$
$$-x_5x_9 \geq 0$$
$$1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0$$
$$x_5x_8 - x_6x_7 \geq 0$$
$$1 - (x_1 - x_7)^2 - (x_2 - x_8)^2 \geq 0$$
$$x_9 \geq 0$$
$$1 - x_7^2 - (x_8 - x_9)^2 \geq 0$$

$x_1x_4 - x_2x_3 \geq 0$

(6) Subject to

$105 - 4x_1 - 5x_2 + 3x_7 - 9x_8 \geq 0$

$-10x_1 + 8x_2 + 17x_7 - 2x_8 \geq 0$

$8x_1 - 2x_2 - 5x_9 + 2x_{10} + 12 \geq 0$

$-3(x_1 - 2)^2 - 4(x_2 - 3)^2 - 2x_3^2 + 7x_4 + 120 \geq 0$

$-5x_1^2 - 8x_2 - (x_3 - 6)^2 + 2x_4 + 40 \geq 0$

$-0.5(x_1 - 8)^2 - 2(x_2 - 4)^2 - 3x_5^2 + x_6 + 30 \geq 0$

$-x_1^2 - 2(x_2 - 2)^2 + 2x_1x_2 - 14x_5 + 6x_6 \geq 0$

$3x_1 - 6x_2 - 12(x_9 - 8)^2 + 7x_{10} \geq 0$

(7) Subject to

$g_1(\mathbf{x}) = 35.82 - 0.222x_{10} - bx_9 \geq 0$

$g_2(\mathbf{x}) = -133 + 3x_7 - ax_{10} \geq 0$

$g_3(\mathbf{x}) = -g_1(\mathbf{x}) + x_9(1/b - b) \geq 0$

$g_4(\mathbf{x}) = -g_2(\mathbf{x}) + (1/a - a)x_{10} \geq 0$

$g_5(\mathbf{x}) = 1.12x_1 + 0.13167x_1x_8 - 0.00667x_1x_8^2 - ax_4 \geq 0$

$g_6(\mathbf{x}) = 57.425 + 1.098x_8 - 0.038x_8^2 + 0.325x_6 - ax_7 \geq 0$

$g_7(\mathbf{x}) = -g_5(\mathbf{x}) + (1/a - a)x_4 \geq 0$

$g_8(\mathbf{x}) = -g_6(\mathbf{x}) + (1/a - a)x_7 \geq 0$

$a = 0.99$, $b = 0.9$

(8) Subject to

$x_3 - x_2 \geq 0$

$x_2 - x_1 \geq 0$

$1 - 0.002x_7 + 0.002x_8 \geq 0$

$x_{13} - 1.262626x_{10} + 1.231059x_3x_{10} \geq 0$

$x_5 - 0.03475x_2 - 0.975x_2x_5 + 0.00975x_2^2 \geq 0$

$x_6 - 0.03475x_3 - 0.975x_3x_6 + 0.00975x_3^2 \geq 0$

$x_5x_7 - x_1x_8 - x_4x_7 + x_4x_8 \geq 0$

$1 - 0.002(x_2x_9 + x_5x_8 - x_1x_8 - x_6x_9) - x_5 - x_6 \geq 0$

$x_2x_9 - x_3x_{10} - x_6x_9 - 500x_2 + 500x_6 + x_2x_{10} \geq 0$

$x_2 - 0.9 - 0.002(x_2x_{10} - x_3x_{10}) \geq 0$

$x_4 - 0.03475x_1 - 0.975x_1x_4 + 0.00975x_1^2 \geq 0$

$x_{11} - 1.262626x_8 + 1.231059x_1x_8 \geq 0$

$x_{12} - 1.262626x_9 + 1.231059x_2x_9 \geq 0$