Environ Geol (2008) 56:255–267
DOI 10.1007/s00254-007-1161-4

ORIGINAL ARTICLE

Slope stability analysis: a support vector machine approach

Pijush Samui

Received: 28 September 2007 / Accepted: 10 December 2007 / Published online: 23 February 2008

Abstract Artificial Neural Network (ANN) such as backpropagation learning algorithm has been successfully used in slope stability problem. However, generalization ability of conventional ANN has some limitations. For this reason, Support Vector Machine (SVM) which is firmly based on the theory of statistical learning has been used in slope stability problem. An interesting property of this approach is that it is an approximate implementation of a structural risk minimization (SRM) induction principle that aims at minimizing a bound on the generalization error of a model, rather than minimizing only the mean square error over the data set. In this study, SVM predicts the factor of safety that has been modeled as a regression problem and stability status that has been modeled as a classification problem. For factor of safety prediction, SVM model gives better result than previously published result of ANN model. In case of stability status, SVM gives an accuracy of 85.71%.

Keywords Artificial Neural Network · Slope stability · Support Vector Machine

Introduction

Limit equilibrium technique based upon the methods of slices was used to determine the stability of slope (Fellenius 1936; Bishop 1955; Bishop and Morgenstern 1960; Morgenstern and Price 1965). The results obtained from this method is comparable with finite and boundary element methods. This method can easily handle the complex slope geometry, effect of pore water pressure, layered soil and effect of pseudo static earthquake body forces. However, a major disadvantage of this method is that it does not address the issue of kinematics. In upper bound theorem of limit analysis, the kinematic admissibility of the chosen collapse mechanism has always been guaranteed. A number of publications exist about the application of upper bound theorem of limit analysis in slope stability problem (Chen et al. 1969; Karal 1977a, 1977b; Chen and Liu 1990; Michalowski 1995, 2002).

Recently, Artificial Neural Network (ANN) has been successfully used in slope stability problem (Sakellatiou and Ferentinou 2005; Samui and Kumar 2006), but ANN has some limitations. The limitations are listed below:

- A major disadvantage of ANN models is that, unlike other statistical models, they provide no information about the relative importance of the various parameters (Park and Rilett 1999).
- In ANN, as the knowledge acquired during training is stored in an implicit manner, it is very difficult to come up with reasonable interpretation of the overall structure of the network (Kecman 2001). This lead to the term "black box" which many researchers use while referring to ANN's behavior.
- In addition, ANN has some inherent drawbacks such as slow convergence speed, less generalizing performance, arriving at local minimum and over-fitting problems.

With an objective to study the stability of slope, Support Vector Machine (SVM) has been used to predict the factor of safety that has been modeled as a regression problem and stability status that has been modeled as a classification problem. The SVM based on statistical learning theory has

P. Samui (✉)
Department of Civil Engineering, Indian Institute of Science, Bangalore 560 012, India
e-mail: pijush.phd@gmil.com

been developed by Vapnik (1995). It provides a new, efficient novel approach to improve the generalization performance and can attain a global minimum. In general, SVMs have been used for pattern recognition problems. Recently it has been used to solve non-linear regression estimation and time series prediction by introducing ε-insensitive loss function (Mukherjee et al. 1997; Muller et al. 1997; Vapnik 1995; Vapnik et al. 1997). The SVM implements the structural risk minimization principle (SRMP), which has been shown to be superior to the more traditional Empirical Risk Minimization Principle (ERMP) employed by many of the other modeling techniques (Osuna et al. 1997). SRMP minimizes an upper bound of the generalization error, whereas ERMP minimizes the training error. In this way, SRMP produces better generalization than traditional techniques. Another major advantage of the SVM is its optimization algorithm, which includes solving a linearly constrained quadratic programming function leading to a unique, optimal, and global solution compared to the ANN. In SVM, the number of support vectors has determined by algorithm rather than by trial-and-error which has been used by ANN for determining the number of hidden nodes. This study uses the database collected by Sakellariou and Ferentinou (2005) (Table 1). The dataset consists the magnitude of unit weight (γ), cohesion (c), angle of internal friction (ϕ), slope angle (β), height (H), pore water pressure coefficient (r_u), factor of safety (FS) and status of slope (S) i.e. stable or failed.

Support vector classification

The SVM has recently emerged as an elegant pattern recognition tool and a better alternative to ANN methods. The method has been developed by Vapnik (1995) and is gaining popularity due to many attractive features. This section of the paper serves as an introduction to this relatively new technique. Details of this method can be found in Boser et al. 1992, Cortes and Vapnik (1995), Gualtieri et al. (1999) and Vapnik (1998). A binary classification problem is considered having a set of training vectors (D) belonging to two separate classes.

$$\mathrm{D} = \left\{\left(x^1, y^1\right), \ldots\ldots\ldots\ldots, \left(x^1, y^1\right)\right\} \quad x \in R^n,\ y \in \{-1, +1\} \tag{1}$$

Where $x \in R^n$ is an n-dimensional data vector with each sample belonging to either of two classes labeled as $y \in \{-1, +1\}$, and l is the number of training data. The main aim is to find a generalized classifier that can distinguish the two classes $(-1, +1)$ from the set of the

Table 1 Dataset used in this study

γ (kN/m^3)	c (kPa)	ϕ (°)	β (°)	H (m)	r_u	FS	S
18.68	26.34	15	35	8.23	0	1.11	Failed
16.5	11.49	0	30	3.66	0	1	Failed
18.84	14.36	25	20	30.5	0	1.875	Stable
18.84	57.46	20	20	30.5	0	2.045	Stable
28.44	29.42	35	35	100	0	1.78	Stable
28.44	39.23	38	35	100	0	1.99	Stable
20.6	16.28	26.5	30	40	0	1.25	Failed
14.8	0	17	20	50	0	1.13	Failed
14	11.97	26	30	88	0	1.02	Failed
25	120	45	53	120	0	1.3	Stable
26	150.05	45	50	200	0	1.2	Stable
18.5	25	0	30	6	0	1.09	Failed
18.5	12	0	30	6	0	0.78	Failed
22.4	10	35	30	10	0	2	Stable
21.1	10	30.34	30	20	0	1.7	Stable
22	20	36	45	50	0	1.02	Failed
22	0	36	45	50	0	0.89	Failed
12	0	30	35	4	0	1.46	Stable
12	0	30	45	8	0	0.8	Failed
12	0	30	35	4	0	1.44	Stable
12	0	30	45	8	0	0.86	Failed
23.47	0	32	37	214	0	1.08	Failed
16	70	20	40	115	0	1.11	Failed
20.41	24.9	13	22	10.67	0.35	1.4	Stable
19.63	11.97	20	22	12.19	0.405	1.35	Failed
21.82	8.62	32	28	12.8	0.49	1.03	Failed
20.41	33.52	11	16	45.72	0.2	1.28	Failed
18.84	15.32	30	25	10.67	0.38	1.63	Stable
18.84	0	20	20	7.62	0.45	1.05	Failed
21.43	0	20	20	61	0.5	1.03	Failed
19.06	11.71	28	35	21	0.11	1.09	Failed
18.84	14.36	25	20	30.5	0.45	1.11	Failed
21.51	6.94	30	31	76.81	0.38	1.01	Failed
14	11.97	26	30	88	0.45	0.625	Failed
18	24	30.15	45	20	0.12	1.12	Failed
23	0	20	20	100	0.3	1.2	Failed
22.4	100	45	45	15	0.25	1.8	Stable
22.4	10	35	45	10	0.4	0.9	Failed
20	20	36	45	50	0.25	0.96	Failed
20	20	36	45	50	0.5	0.83	Failed
20	0	36	45	50	0.25	0.79	Failed
20	0	36	45	50	0.5	0.67	Failed
22	0	40	33	8	0.35	1.45	Stable
24	0	40	33	8	0.3	1.58	Stable
20	0	24.5	20	8	0.35	1.37	Stable
18	5	30	20	8	0.3	2.05	Stable

training vectors mentioned earlier (D) and also can classify equally well the unseen data. In the current context of classifying slope failure, the two classes labeled as (−1, + 1) may mean failed slope and stable slope. The stability of slope depends on geometry of slope and strength of soil. Strength of soil depends on cohesion (c), angel of internal friction (ϕ), pore water pressure coefficient (r_u) and unit weight of soil (γ). Geometry of slope depends on slope angle (β) and height of slope (H). According to Michalowski (1994), for the determination of stability of homogeneous slope, the aforementioned six parameters (c, ϕ, r_u, γ, β, and H) have to be considered.

In this study, γ, c, ϕ, β, H and r_u are used as input parameters. So, $x = [\gamma, H, c, r_u, \beta, \varphi\ ]$. For a set of data, this would mean a linear hyper plane defined by Eq. (2) which can distinguish the two classes.

$$f(x) = w \cdot x + b = 0 \tag{2}$$

where $w \in R^n$ determines the orientation of a discriminating hyperplane, $b \in R$ is a bias. An example of hyperplane is shown in Fig. 1. For the linearly separable case, a separating hyperplane can be defined for the two classes as

$$\begin{array}{ll} w \cdot x_i + b \geq 1 & (\text{for } y_i = 1) \rightarrow \text{stable slope} \\ w \cdot x_i + b \leq -1 & (\text{for } y_i = -1) \rightarrow \text{failed slope} \end{array} \tag{3}$$

The above two equations can be combined as

$$y_i(w \cdot x_i + b) \geq 1 \tag{4}$$

Sometimes, due to the noise or mixture of classes introduced during the selection of training data, variables $\xi_i > 0$, called slack variables, are used due to the effects of misclassification. So the Eq. (4) can be written as

$$y_i(w \cdot x_i + b) \geq 1 - \xi_i \tag{5}$$

The perpendicular distance from the origin to the plane $w \cdot x_i + b = -1$ is $\frac{|1+b|}{\|w\|}$. Similarly, the perpendicular distance from the origin to the plane $w \cdot x_i + b = 1$ is $\frac{|b-1|}{\|w\|}$ The margin (ρ (w,b)) between the planes is simply

$$\rho(w, b) = \frac{2}{\|w\|} \tag{6}$$

The optimal hyperplane is located where the margin between two classes of interest is maximized (Fig. 2) and the error is minimized. The maximization of this margin leads to the following constrained optimization problem

$$\begin{array}{ll} \text{Minimize :} & \frac{1}{2}\|w\|^2 + C\sum_{i=1}^{l} \xi_i \\ \text{Subjected to :} & y_i(w \cdot x_i + b) \geq 1 - \xi_i \end{array} \tag{7}$$

The constant $0 < C < \infty$, a parameter defines the trade-off between the number of misclassifications in the training data and the maximization of margin. A large C assigns higher penalties to errors so that the SVM is trained to minimize error with lower generalization, whereas a small C assigns fewer penalties to errors; this allows the minimization of margin with errors, thus higher generalization ability. If C goes to infinitely large, SVM would not allow the occurrence of any error and result in a complex model, whereas when C goes to zero, the result would tolerate a large amount of errors and the model would be less complex.

In order to solve the above optimization problem (7), the Lagrangian is constructed as follows:

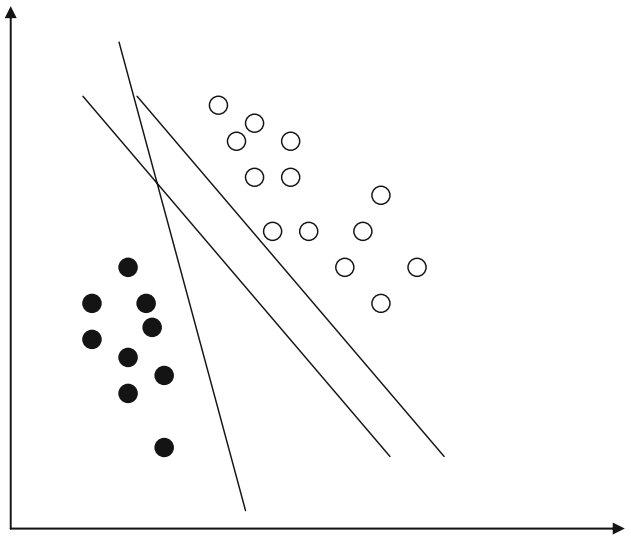

Fig. 1 An example of hyperplane

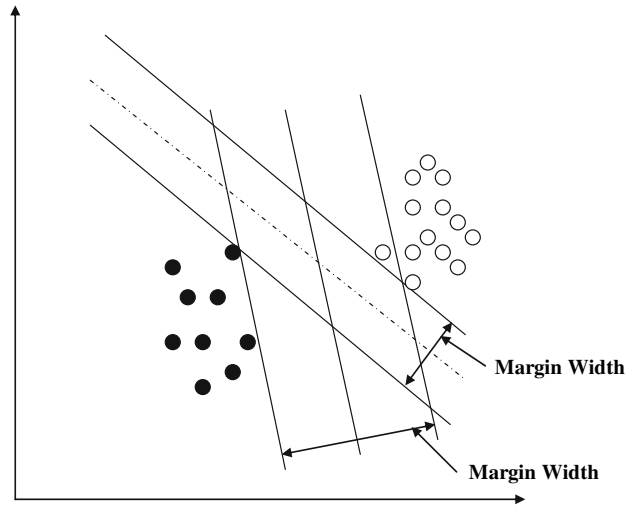

Fig. 2 Margin width of different hyperplanes

$$L(\mathrm{w}, b, \alpha, \beta, \xi) = \frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{1} \xi_i\right) - \sum_{i=1}^{1} \alpha_i\{[(w \cdot x_i + b)]y_i - 1 + \xi_i\} - \sum_{i=1}^{1} \beta_i \xi_i \quad (8)$$

Where α, β are the Lagrange multipliers. The solution to the constrained optimization problem is determined by the saddle point of the Lagrangian function L (w, b, α, β, xi), which has to be minimized with respect to w, b and ξ. Thus, differentiating L (w, b, α, β, ξ) with respect to w, b and ξ and setting the results equal to zero, the following three conditions have been obtained:

$$\text{Condition 1}: \quad \frac{\partial L(w, b, \alpha, \beta, \xi)}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{1} \alpha_i y_i x_i$$

$$\text{Condition 2}: \quad \frac{\partial L(w, b, \alpha, \beta, \xi)}{\partial b} = 0 \Rightarrow \sum_{i=1}^{1} \alpha_i y_i = 0$$

$$\text{Condition 3}: \quad \frac{\partial L(w, b, \alpha, \beta, \xi)}{\partial \xi} = 0 \Rightarrow \alpha_{\mathrm{i}} + \beta_{\mathrm{i}} = C \quad (9)$$

Hence from equations 8, 9 the equivalent optimization problem becomes (Osuna et al, 1997),

$$\text{Maximize}: \quad \sum_{i=1}^{1} \alpha_i - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{1} \alpha_i \alpha_j y_i y_j (x_i \cdot x_j)$$

Subjected to :

$$\sum_{i=1}^{1} \alpha_i y_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le C, \quad \text{for } i = 1, 2, \ldots, 1 \quad (10)$$

Solving Eq. (10) with constraints determines the Lagrange multipliers. According to the Karush–Kuhn–Tucker (KKT) optimality condition (Fletcher 1987), some of the multipliers will be zero. The non-zero multipliers are called support vectors (see Fig. 3). In conceptual terms, the support vectors are those data points that lie closest to the optimal hyperplane and are therefore the most difficult to classify. The value of w and b are calculated from $w = \sum_{i=1}^{l} y_i \alpha_i x_i$ and $b = -\frac{1}{2} w[x_{+1} + x_{-1}]$, where x_{+1} and x_{-1} are the support vectors of class labels +1(stable slope) and −1 (failed slope), respectively. The classifier can then be constructed as:

$$f(x) = \mathrm{sign}(w \cdot x + b) \quad (11)$$

Where sign $(\cdot)$ is the signum function. It gives +1 (stable slope) if the element is greater than or equal to zero and −1 (failed slope) if it is less than zero.

In case where linear supporting hyper plane is inappropriate, SVM maps input data into a high-dimensional feature space through some non-linear mapping (Boser et al. 1992) (see Fig. 4). This method easily converts a linear classification learning algorithm into a non-linear one, by mapping the original observations into a higher-dimensional non-linear space so that linear classification in the new space is equivalent to non-linear classification in the original space. After replacing x by its mapping in the feature space ($\Phi(x)$), the optimization problem of Eq. (11) becomes

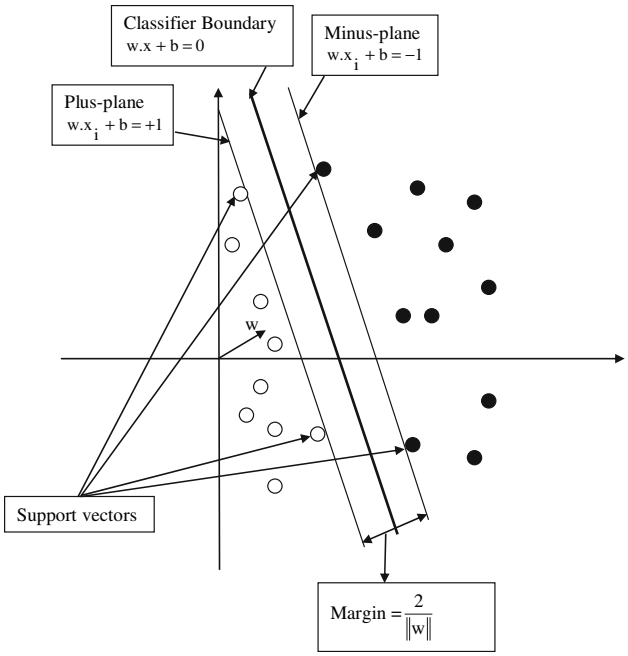

Fig. 3 Support vectors with maximum margin

$$\text{Maximize}: \quad \sum_{i=1}^{1} \alpha_i - \frac{1}{2}\sum_{i=1}^{1}\sum_{j=1}^{l} \alpha_i \alpha_j y_i y_j \left(\Phi(x_i) \cdot \Phi(x_j)\right)$$

Subjected to :

$$\sum_{i=1}^{1} \alpha_i y_i = 0 \quad \text{and} \quad 0 \le \alpha_i \le C, \quad \text{for } i = 1, 2, \ldots, 1 \quad (12)$$

Kernel function $K(x_i \cdot x_j) = \Phi(x_i) \cdot \Phi(x_j)$ has been introduced instead of feature space ($\Phi(x)$) to reduce

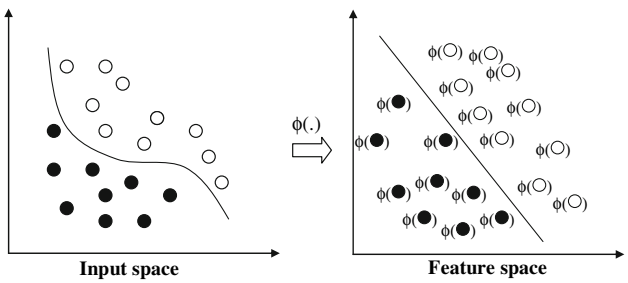

Fig. 4 Concept of non-linear SVM for classification problem

computational demand (Cortes and Vapnik 1995; Cristianini and Shawe-Taylor 2000). Polynomial, radial basis functions and certain sigmoid functions has been used as kernel functions. To get Eq. (11), same procedures have been applied as in linear case.

The main scope of this work is to implement the above classification model for forecasting the status of stability in the case of soil slope. In carrying out the formulation, the data have been divided into two sub-sets: such as

a. A training dataset: this is required to construct the model. In this study, 32 out of the 46 cases of slope are considered for training the dataset.
b. A testing dataset: this is required to estimate the model performance. In this study, the remaining 14 data is considered as testing dataset. The data are normalized against their maximum values (Sincero 2003). The success of SVM to classification problem largely depends on C, type of kernel and its parameters. To train the SVM model, three types of kernel function have been used: They are

    1. Polynomial
    2. Radial basis function
    3. Spline

The value of C has been determined by trial-and-error method. The program is constructed using MATLAB (MathWork Inc. 1999).

Support Vector Regression

The SVM can also be applied to regression problem by the introduction of an alternative loss function (Smola 1996). In this section, a brief introduction on how to construct SVM for regression problem is presented. More details can be found in many publications (Smola 1996; Dibike et al. 2001; Smola and Scholkopf 2004; Khan and Coulibaly 2006). This study uses the SVM as a regression technique by introducing a ε-insensitive loss function. There are three distinct characteristics of SVMs when they are used to estimate the regression function. First of all, SVMs estimate the regression using a set of linear functions that are defined in a high dimensional space. Secondly, SVMs carry out the regression estimation by risk minimization where the risk is measured using Vapnik's ε-insensitive loss function. Thirdly, SVMs use a risk function consisting of the empirical error and a regularization term which is derived from the structural risk minimization (SRM) principle. In SVM, high generalization performance is achieved by minimizing the sum of the training set error and a term that depends on the Vapnik–Chervonenkis (VC) dimension. This study uses the SVM as a regression technique by introducing an ε-insensitive loss function. The ε-insensitive loss function ($L_\varepsilon(y)$) can be described in the following way:

$$L_\varepsilon(y) = 0 \quad \text{for} \quad |f(x) - y| < \varepsilon \quad \text{otherwise} \quad L_\varepsilon(y) = |f(x) - y| - \varepsilon \tag{13}$$

This defines an ε tube (Fig. 5) so that if the predicted value is within the tube the loss is zero, while if the predicted point is outside the tube, the loss is the magnitude of the difference between the predicted value and the radius, ε, of the tube. Assume that the training dataset consists of one training sample $\{(x_1, y_1), \ldots (x_l, y_l)\}$ where x is the input and y is the output. A problem with learning is related to choosing a function that predicts the actual response y as closely as possible, with a precision of ε. Let us assume a linear function

$$f(x) = (w \cdot x) + b \tag{14}$$

where, $w \in R^n$ and $b \in r$; w = an adjustable weight vector; b = the scalar threshold; R^n = n-dimensional vector space; and r = one dimensional vector space. In this study, x = [γ, H, c, r_u, β, φ] and y = [FS].

The main aim of SVMs is to find a function $f(x)$ that gives a deviation ε from the actual output (y), which is, at

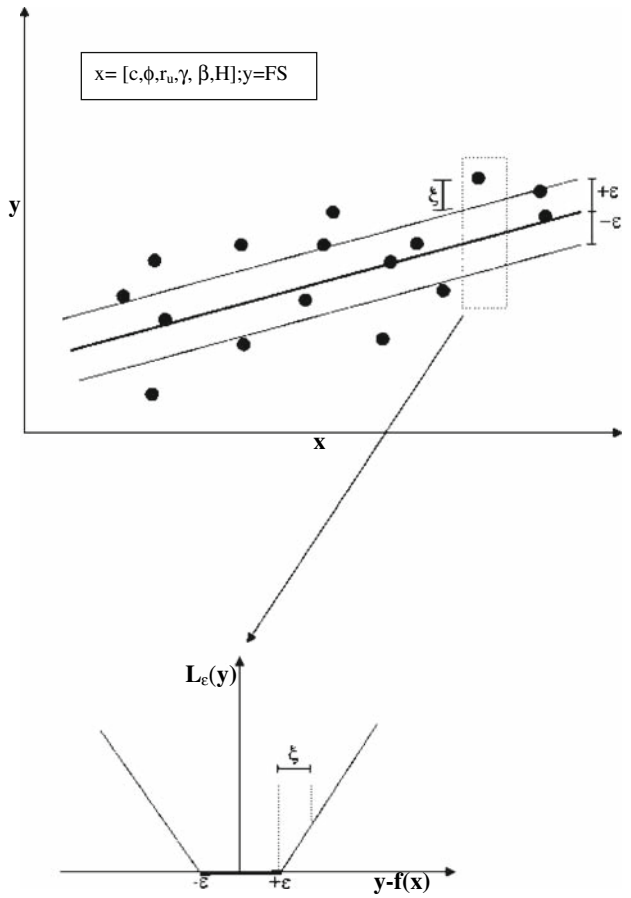

Fig. 5 Prespecified accuracy ε and slack variable ξ in support vector regression (Scholkopf 1997)

the same time, at the same time as flat as possible. Flatness in the case of Eq. (2) means that one seeks a small w. One way of obtaining this is by minimising the Euclidean norm $\|w\|^2$ (Smola and Scholkopf 2004). The convex optimization problem thus involves:

$$\begin{aligned} &\text{Minimize :} && \tfrac{1}{2}\|w\|^2 \\ &\text{Subjected to :} && y_i - (\langle w \cdot x_i \rangle + b) \le \varepsilon, \quad i = 1,2,\ldots,1 \\ & && (\langle w \cdot x_i \rangle + b) - y_i \le \varepsilon, \quad i = 1,2,\ldots,1 \end{aligned} \tag{15}$$

The best regression line is defined by minimizing the following cost function

$$\begin{aligned} &\text{Minimize :} && \tfrac{1}{2}\|w\|^2 + C\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right) \\ &\text{Subjected to :} && y_i - (\langle w \cdot x_i \rangle + b) \le \varepsilon + \xi_i, \quad i = 1,2,\ldots,1 \\ & && (\langle w \cdot x_i \rangle + b) - y_i \le \varepsilon + \xi_i^*, \quad i = 1,2,\ldots,1 \\ & && \xi_i \ge 0 \text{ and } \quad \xi_i^* \ge 0, \qquad\qquad i = 1,2,\ldots,1 \end{aligned} \tag{16}$$

The above optimization problem, Eq. (16), consists of a 2-norm penalty associated with the regression coefficients, an error term multiplied by the error weight, C, and a set of constraints. Using this cost function, the goal is to simultaneously minimize both the coefficient's size and the prediction errors. This is important because large coefficients might hamper generalization because these can cause excessive variance. This is an approach which is often used in multivariate calibration and that can deal with ill-posed problems. These can occur when performing spectral calibration (because the number of objects is smaller than the number of variables or the latter are correlated). In the error term, the prediction errors are penalized linearly with the exception of a deviation of $\pm\varepsilon$. Predictions deviating by more than $+\varepsilon$ or less than $-\varepsilon$ are taken into account by the so-called slack variables indicated by ξ_{i} and ξ_{i}^*, respectively (Fig. 5). This value needs to be optimized by the user. Note that the value of ε does not indicate the desired final prediction error of the model but is a characteristic of the prediction error penalty. In principle, it is possible to use an ε-value of zero. In this case, the error penalty reduces to a regular minimization of absolute values (minimizing the 1-norm of the error). C represents the penalty weight and, in addition to the value of ε, this value also needs to be optimized by the user. If its value is very high, deviations from $\pm\varepsilon$ count heavier in the cost function. For an infinite value of C, a solution is considered best if its error is minimal, even though the regression coefficients size is very high. If an extremely low value of C is chosen, the best result is determined exclusively by the size of the regression weights. As stated above, the value of ε also needs to be defined by the user and it is data and problem-dependent. For example, if noise is present in the data, a larger ε can guide the solutions to be more independent of existing noise. On the other hand, too large values of ε lead to the situation in which no proper predictions can be made. This is caused by the fact that objects with prediction errors larger than $\pm\varepsilon$ are the so-called support vectors, and only these determine the final prediction of the SVM model. This is explained later. 'ε' is a trade-off between the sparseness of the representation and closeness to the data. Hence, the Lagrangian function is constructed from both the objective function and corresponding constraints in Eq. (16) as follows:

$$\begin{aligned} L(w,\xi,\xi^*,\alpha,\alpha^*,\gamma,\gamma^*) = \frac{\|w\|^2}{2} + C\left(\sum_{i=1}^{l}\left(\xi_i + \xi_i^*\right)\right) \\ - \sum_{i=1}^{l}\alpha_i\left[\varepsilon + \xi_i - y_i + \langle w \cdot x_i \rangle + b\right] \\ - \sum_{i=1}^{l}\alpha_i^*\left[\varepsilon + \xi_i^* + y_i - \langle w \cdot x_i \rangle - b\right] \\ - \sum_{i=1}^{l}\left(\gamma_i\xi_i + \gamma_i^*\xi_i^*\right) \end{aligned} \tag{17}$$

where L is the Lagrangian and α, α^*, γ and γ^* are the Lagrangian multipliers. The partial derivatives of L with respect to ω, b, ξ, and ξ^* have to be vanished to satisfy the saddle point condition.

$$\begin{aligned} &\partial_w L = 0 \Rightarrow w = \sum_{i=1}^{l} x_i\left(\alpha_i - \alpha_i^*\right) \\ &\partial_b L = 0 \Rightarrow \sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^* \\ &\partial_\xi L = 0 \Rightarrow \sum_{i=1}^{l}\gamma_i = \sum_{i=1}^{l}(C - \alpha_i) \\ &\partial_{\xi^*} L = 0 \Rightarrow \sum_{i=1}^{l}\gamma_i^* = \sum_{i=1}^{l}\left(C - \alpha_i^*\right) \end{aligned} \tag{18}$$

Substituting (18) into (19) yields the dual optimization problem

$$\begin{aligned} &\text{Maximize :} && -\varepsilon\sum_{i=1}^{l}\left(\alpha_i^* + \alpha_i\right) + \sum_{i=1}^{l} y_i\left(\alpha_i^* - \alpha_i\right) \\ & && -\frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}\left(\alpha_i^* - \alpha_i\right)\left(\alpha_j^* - \alpha_j\right)\left(x_i \cdot x_j\right) \\ &\text{Subjected to :} && \sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^*; 0 \le \alpha_i^* \le C \quad \text{and} \quad 0 \le \alpha_i \le C \end{aligned} \tag{19}$$

The coefficients α_i, α_i^* are determined by solving the above optimization problem (Eq. 19). An important aspect is that some Lagrange multipliers (α_i, α_i^*) will be zero, implying that these training objects are considered to be irrelevant for the final solution (sparseness). The training objects with non-zero Lagrange multipliers are called support vectors. These are the objects with prediction errors larger than $\pm\varepsilon$. In this way, the value of ε determines the amount of support vectors. Obviously, if ε is too large, too few support vectors are selected, which leads to a decrease in the final prediction performance. Furthermore, the Lagrange multipliers of the support vectors all have different values, which mean that one support vector is considered to be more important than another one. So Eq. (14) can be written as:

$$f(x) = \sum_{\text{support vectors}} (\alpha_i - \alpha_i^*)(x_i \cdot x) + b$$
$$\text{Where} \quad b = -\left(\frac{1}{2}\right) w \cdot [x_r + x_s] \tag{20}$$

From Eq. (20) it is clear that w has been completely described as a linear combination of training patterns. So, the complexity of a function represented by support vectors is independent of the dimensionality of input space, and it depends only on the number of support vectors. The entry of the data in inner products is very important because: (1) the dimension of the objects does not appear in the problem to be solved and (2) extension of this linear approach to non-linear regression can be easily made.

When linear regression is not appropriate, then input data must be mapped into a high-dimensional feature space through some non-linear mapping (Boser et al. 1992). In optimization problem expressed in Eq. (19), x has been replaced by the feature space, $\Phi(x)$ (see Fig. 6). So, the optimization problem (Eq. 19) can be written as:

$$\text{Maximize}: \quad -\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right)\left(\Phi(x_i)\cdot\Phi(x_j)\right)$$
$$\text{Subjected to}: \quad \sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^*; 0 \le \alpha_i^* \le C \quad \text{and} \quad 0 \le \alpha_i \le C \tag{21}$$

The concept of kernel function [$K(x_i, x_j) = \Phi(x_i) \cdot \Phi(x_j)$] has been introduced to reduce the computational demand (Cortes and Vapnik 1995; Cristianini and Shawe-Taylor 2000). So optimization problem can be written as:

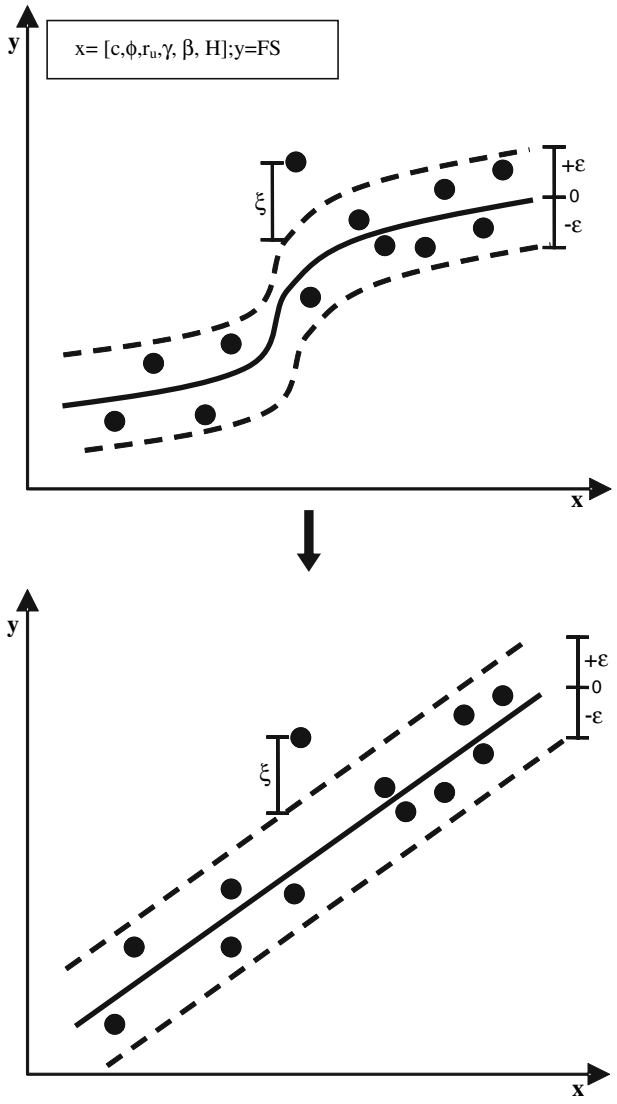

Fig. 6 Concept of non-linear regression

$$\text{Maximize}: \quad -\varepsilon\sum_{i=1}^{l}(\alpha_i^* + \alpha_i) + \sum_{i=1}^{l} y_i(\alpha_i^* - \alpha_i) - \frac{1}{2}\sum_{i=1}^{l}\sum_{j=1}^{l}(\alpha_i^* - \alpha_i)\left(\alpha_j^* - \alpha_j\right)\left(K(x_i \cdot x_j)\right)$$
$$\text{Subjected to}: \quad \sum_{i=1}^{l}\alpha_i = \sum_{i=1}^{l}\alpha_i^* \quad 0 \le \alpha_i^* \le C \quad \text{and} \quad 0 \le \alpha_i \le C \tag{22}$$

The introduction of kernels according to Mercer's theorem (Vapnik 1995) avoids an explicit formation of the non-linear mapping, makes the dimension of feature space even infinite, and reduces the computational load greatly by enabling the operation in low-dimensional input space instead of high-dimensional feature space. Some common

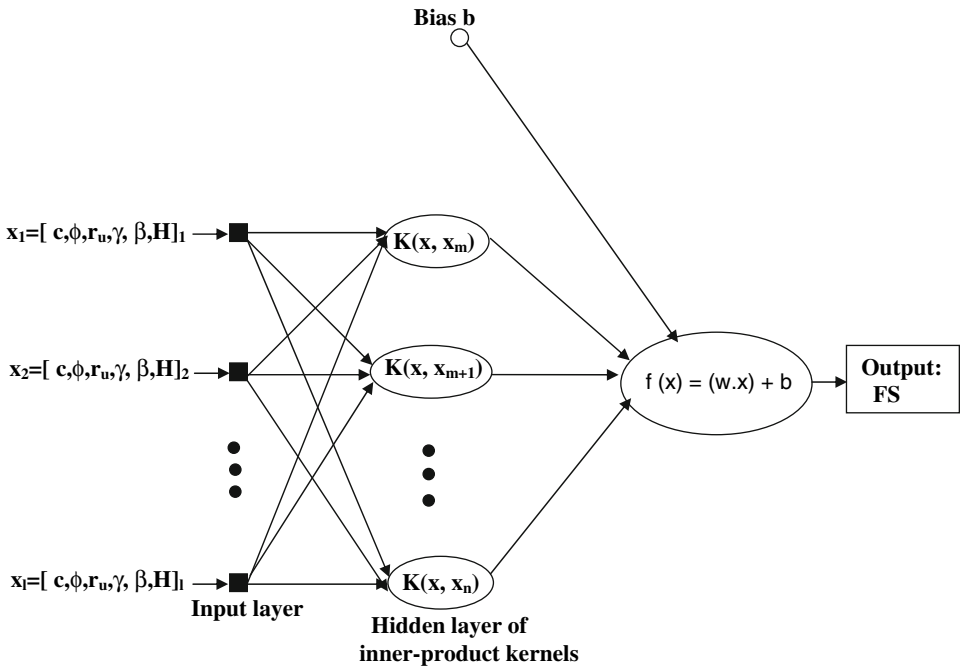

Fig. 7 SVM architecture for FS prediction

kernels have been used, such as a polynomial (homogeneous), polynomial (non-homogeneous), radial basis function, Gaussian function, sigmoid for non-linear cases. The Kernel representation offers a powerful alternative by using linear machines in hypothesizing complex real-world problems as opposed to ANN- based learning paradigms, which use multiple layers of threshold linear functions (Cristianini and Shawe-Taylor 2000). The regression function Eq. (13) has been obtained by applying the same procedure as in the linear case. An important characteristic of this optimization problem is that the solution is global and deterministic (i.e. given the same training set and values of ε and C, the same solution is always found; that is, no stochastic events are present during the building of the model), which is in contrast to ANNs. Figure 7 shows the architecture of an SVM for FS prediction.

In the present study, SVM has been used for the prediction of FS for the same dataset of slope. ε-insensitive loss function has been used in this analysis. In SVM modeling for regression problem, the data hae been divided into two sub-sets; such as

(a) A training dataset: this is required to construct the model.

(b) A testing dataset: this is required to estimate the model performance.

The same training, testing dataset and normalization technique (as used in classification model) has been used. In this model, γ, c, ϕ, β, H and r_u are used as input parameters, while FS value is the output from this model. When applying SVM, in addition to the specific kernel parameters, the optimum values of the capacity factor C and the size of the error-insensitive zone ε should be determined during the modeling experiment. In this study, spline, radial basis function and polynomial function are used as the kernel function of the SVM. In the present study, training and testing of SVM for regression problem has been carried out by using MATLAB (MathWork Inc 1999).

Results and discussions

The design C value of classification model for each kernel is shown in Table 2 along with training performance, testing performance and number of support vectors. Training and testing performance have been calculated from the following formula

Table 2 General performance of SVM in classification problem using different kernels

Kernel	C	Number of support vectors	Training performance (%)	Testing performance (%)
Polynomial, degree = 2	10	11	100	85.71
Radial basis function, width (σ) = 1	100	14	100	78.57
Spline	100	10	100	78.57

$$\text{Training performance (\%) or Testing performance (\%)} = \left(\frac{\text{No data predicted accurately by SVM}}{\text{Total data}}\right) \times 100 \quad (23)$$

Table 2 shows that, in terms of model accuracy, the performance of polynomial kernel is better compared with other kernels. Polynomial kernel gives an accuracy of 85.71%, with no errors in the training patterns and two errors in testing patterns. The other observation is that spline kernel produces lowest number of support vector. Table 3 shows the performance of different kernel using testing dataset. In this study, SVM model employs approximately 31–44% (for radial basis function = 43.75%, polynomial kernel = 34.37% and spline kernel = 31.25%) of the training data as support vectors for classification problem. It is worth mentioning here that the support vectors in the SVM model represent prototypical examples. The prototypical examples exhibit the essential features of the information content of the data, and thus are able to transform the input data into the specified targets. So, there is real advantage gained in terms of sparsity. Sparseness means that a significant number of the weights are zero (or effectively zero), which has the consequence of producing compact, computationally efficient models, which in addition are simple and therefore produce smooth functions.

In SVM method for regression problem, identification of the optimal values for C and ε for specific kernel is largely a trial-and-error process, which does, however, become much easier with practice. Figures 8, 9, and 10 show the performance of the SVM model for training dataset for polynomial, radial basis function, spline kernel, respectively. SVM training involves solving a uniquely solvable quadratic optimization problem, unlike ANN training, which requires complex non-linear optimization with the danger of getting stuck in local minima.

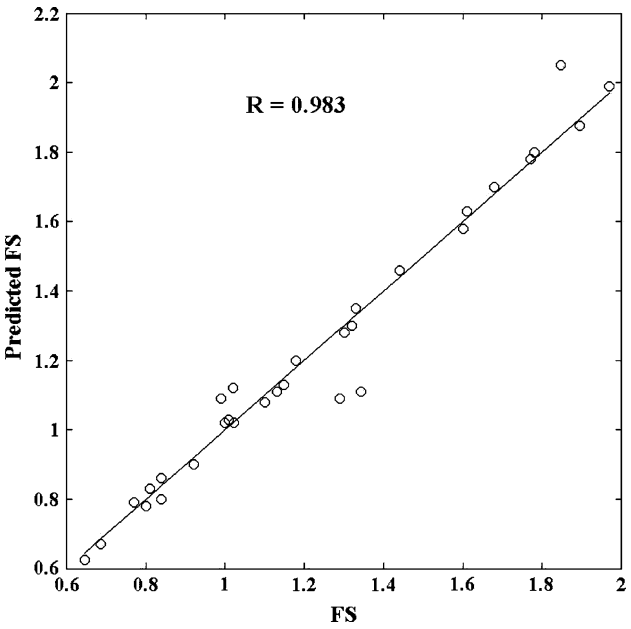

Fig. 8 Performance of SVM model for training dataset using polynomial kernel

In order to evaluate the capabilities of the SVM model for regression problem, the model is validated with new data that are not part of the training dataset. Figures 11, 12, and 13 show the performance of the SVM model for testing dataset for polynomial, radial basis function and spline,

Table 3 Performance of SVM model for classification problem using testing dataset

γ (kN/m^3)	c (kPa)	ϕ (°)	β (°)	H (m)	r_u	Actual class	Predicted class		
							Polynomial	Radial basis function	Spline
16.5	11.49	0	30	3.66	0	−1	−1	−1	−1
18.84	57.46	20	20	30.5	0	1	1	1	1
20.6	16.28	26.5	30	40	0	−1	−1	−1	−1
22.4	10	35	30	10	0	1	1	1	1
22	0	36	45	50	0	−1	−1	−1	−1
12	0	30	35	4	0	1	1	1	1
16	70	20	40	115	0	−1	−1	−1	−1
20.41	24.9	13	22	10.67	0.35	1	−1	−1	−1
21.82	8.62	32	28	12.8	0.49	−1	−1	1	1
18.84	0	20	20	7.62	0.45	−1	−1	−1	−1
21.51	6.94	30	31	76.81	0.38	−1	−1	−1	−1
20	20	36	45	50	0.25	−1	−1	−1	−1
22	0	40	33	8	0.35	1	1	1	1
20	0	24.5	20	8	0.35	1	−1	−1	−1

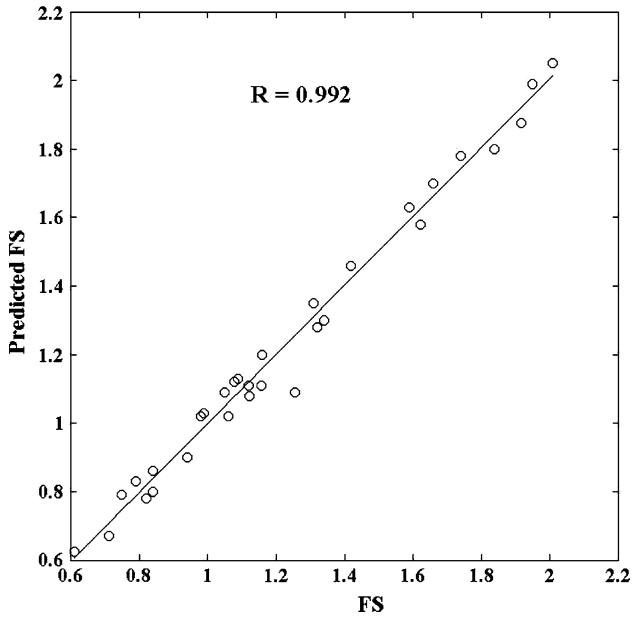

Fig. 9 Performance of SVM model for training dataset using radial basis function kernel

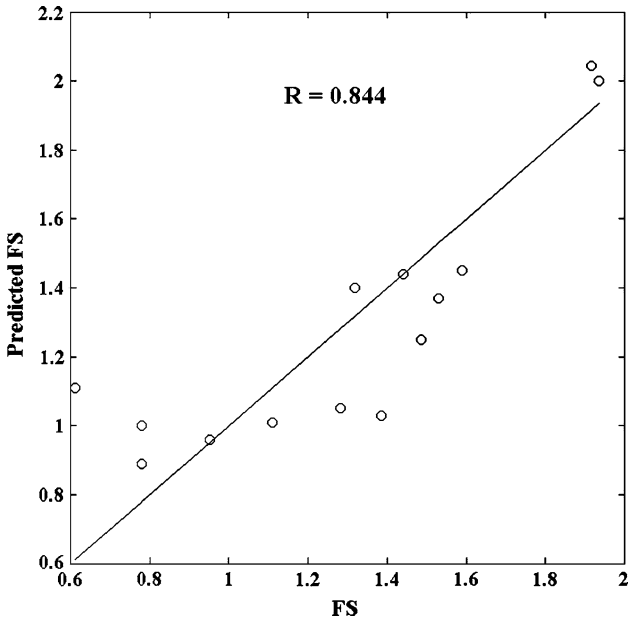

Fig. 11 Performance of SVM model for testing dataset using polynomial kernel

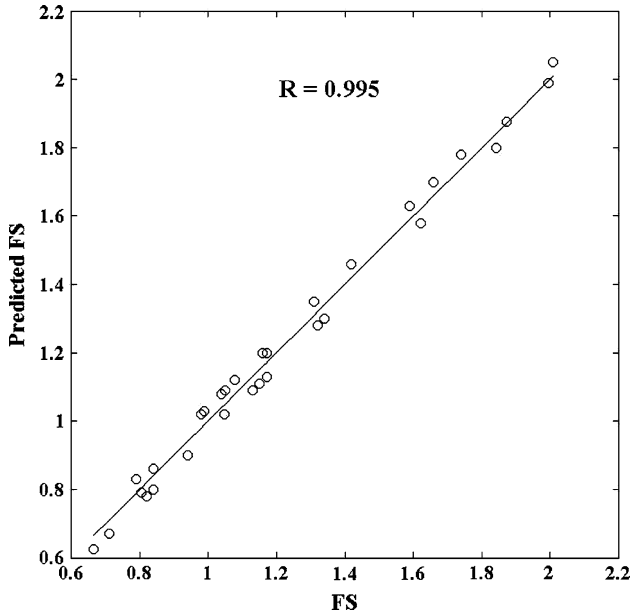

Fig. 10 Performance of SVM model for training dataset using spline kernel

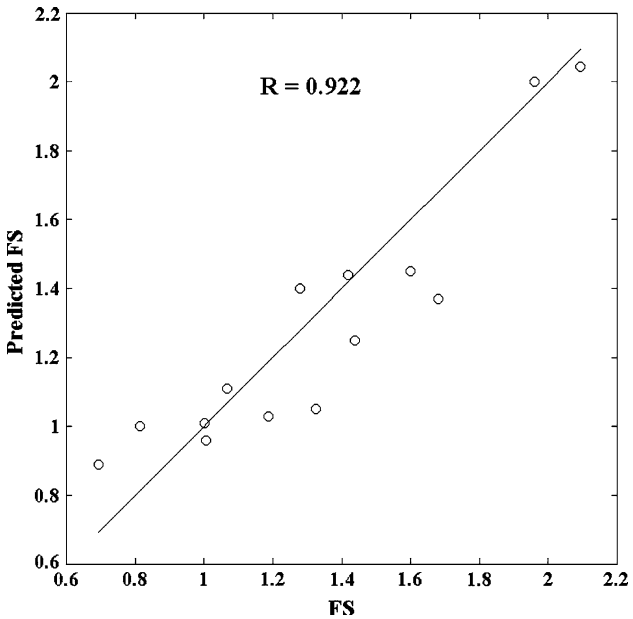

Fig. 12 Performance of SVM model for testing dataset using radial basis function kernel

respectively. The width (σ) of radial basis kernel is 1 and the degree of polynomial kernel is 2. Table 4 shows the coefficient of correlation (R) of SVM for each kernel type with the corresponding numbers of support vectors, C and ε. From the Table 4, it is clear that radial basis function kernel gives better result than other kernels. Spline kernel produces lowest number of support vector. In this study, SVM model employs approximately 81–88% (for radial basis function = 84.37%, polynomial kernel = 87.50% and spline kernel = 81.25%) of the training data as support vectors for regression problem. So, there is very little gained in terms of sparsity.

The loss of performance with respect to the testing set addresses SVM's susceptibility to overtraining. There is a marginal reduction in performance on the testing dataset (i.e. there is a difference between SVM performance on training and testing) for the SVM model (for regression as well as classification problem). So, SVM has the ability to

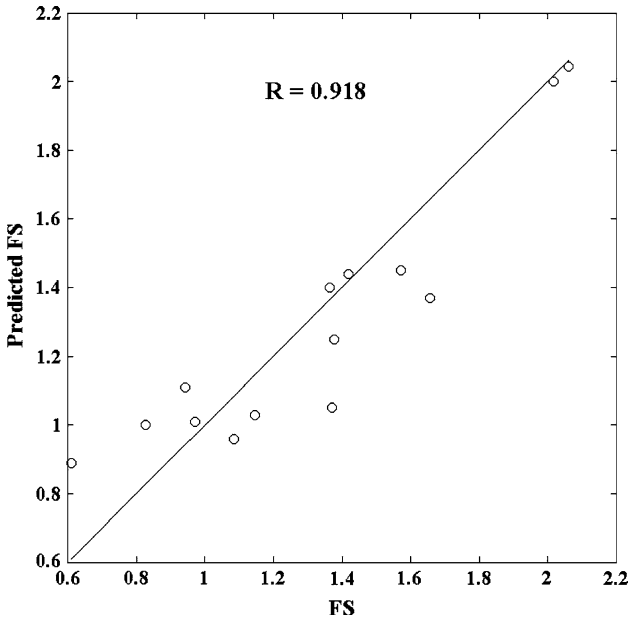

Fig. 13 Performance of SVM model for testing dataset using spline kernel

avoid overtraining, and hence it has good generalization capability. In prediction models, there are infinitely many functions that may provide an accurate fit to the finite testing set. Notwithstanding this, SVM formulation does not try to fit data. Instead, it tries to capture underlying functions from which the data were generated irrespective of the presence of noise. For SVM, this insensitivity to noise in the data is attributed to the ε-insensitive loss function in the model formulation. This feature also provides control over model complexity in ways that alleviate the problems of over- and under-fitting.

In this SVM modeling for regression problem, the data (Table 5) from the models introduced by Hoek and Bray (1981), Lin et al. (1988), Madzic (1988) and Hudson (1992) has been used to examine the generalization capability of developed SVM model, and a comparison has been also done with result of Sakellariou and Ferentinou (2005). Sakellariou and Ferentinou (2005) had used

Table 4 General performance of SVM in regression problem using different kernels

Kernel	C	ε	Number of support vectors	Training performance	Testing performance
Polynomial, degree = 2	20	0.01	28	0.983	0.844
Radial basis function,width (σ) = 1	100	0.02	27	0.992	0.922
Spline	100	0.02	26	0.995	0.918

Table 5 Data from different case histories

Reference	γ (kN/m^3)	c (kPa)	ϕ (°)	β (°)	H (m)	r_u	FS
Hoek and Bray (1981)	21	20	40	40	12	0	1.84
	21	45	25	49	12	0.3	1.53
	21	30	35	40	12	0.4	1.49
	21	35	28	40	12	0.5	1.43
	20	10	29	34	6	0.3	1.34
	20	40	30	30	15	0.3	1.84
	18	45	25	25	14	0.3	2.09
	19	30	35	35	11	0.2	2
	20	40	40	40	10	0.2	2.3
Hudson (1992)	18.85	24.8	21.3	29.2	37	0.5	1.07
	18.85	10.34	21.3	34	37	0.3	1.29
Lin et al. (1988)	18.8	30	10	25	50	0.1	1.4
	18.8	25	10	25	50	0.2	1.18
	18.8	20	10	25	50	0.3	0.97
	19.1	10	10	25	50	0.4	0.65
	18.8	30	20	30	50	0.1	1.46
	18.8	25	20	30	50	0.2	1.21
	18.8	20	20	30	50	0.3	1
	19.1	10	20	30	50	0.4	0.65
Madzie (1988)	22	20	22	20	180	0	1.12
	22	20	22	20	180	0.1	0.99

Artificial Neural Network model trained with backpropagation algorithm. Figures 14 and 15 show root-mean-square-error (RMSE) and mean-absolute-error (MAE) for this data, respectively. So, it is clear from Figs. 14 and 15, that the result from this study (for all kernels) is better than the result from the work of Sakellariou and Ferentinou (2005). SVM uses only two or three parameters for regression problem (radial basis function: σ, C and ε; polynomial kernel: degree of polynomial, C and ε; spline kernel: C and ε) and one or two parameters (radial basis function: σ and C; polynomial kernel: degree of polynomial and C; spline kernel: C) for classification problem. In ANN, there are a larger number of controlling parameters, including the number of hidden layers, number of hidden nodes, learning rate, momentum term, number of training epochs, transfer functions, and weight initialization methods. Obtaining an optimal combination of these parameters is a difficult task as well. The determination of stability of slope is complex problem in geotechnical engineering. For most mathematical models that attempt to solve this problem, the lack of physical understanding is usually supplemented by either simplifying the problem or incorporating several assumptions into the models. In contrast, as shown in this study, SVM uses the data alone to determine the parameters of the model. In this case, there is no need to simplify the problem or to incorporate any

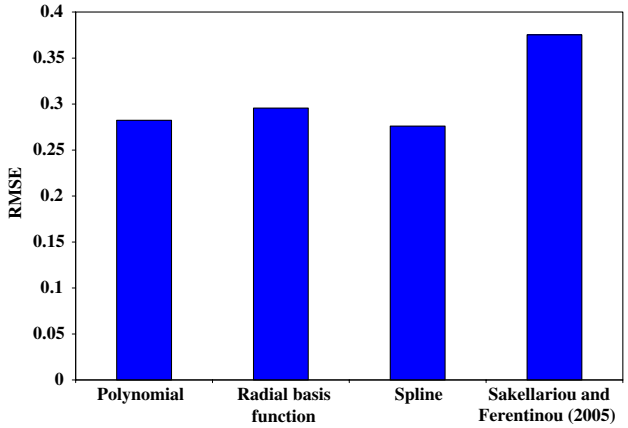

Fig. 14 RMSE for literature data

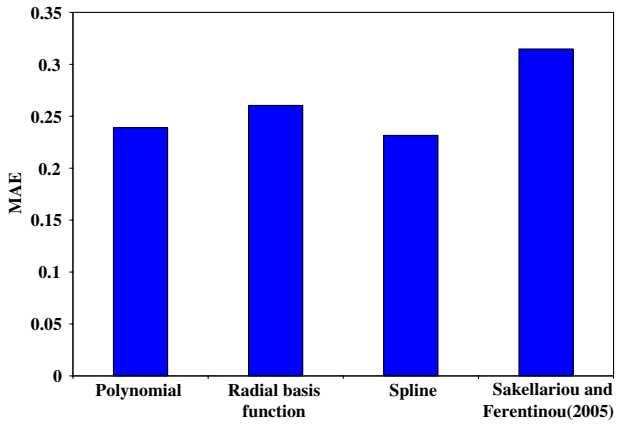

Fig. 15 MAE for literature data

assumptions. Moreover, SVM can always be updated to obtain better results by presenting new training examples as new data become available.

Conclusion

This study suggests that SVM is a powerful computational tool and can be effectively used to slope stability problem. The application of SVM to predict stability status that has been modeled as a classification problem and FS of slope that has been modeled as a regression problem has been described. In both cases, SVM training consists of solving a—uniquely solvable—quadratic optimization problem, which gives always global minimum. In classification problem, SVM gives an accuracy of 87.5%. Advantage of classification method includes simplicity of implementation and relatively low computational cost. For regression problem, SVM gives best result for radial basis function kernel ($R = 0.922$), and the result is also better than the published ANN result. Although SVM gives promising result for both problems, the determination of proper parameters C and ε is still a heuristic process. In general, SVM provides an attractive approach to geological engineer's community.

References

Bishop AW (1955) The use of slip circle in the stability of slopes. Geotechnique 5(1):7–17
Bishop AW, Morgenstern NR (1960) Stability coefficients for earth slopes. Geotechnique 10(4):129–150
Boser BE, Guyon IM, Vapnik VN (1992) A training algorithm for optimal margin classifiers. In: Haussler D (ed) 5th Annual ACM workshop on COLT. ACM, Pittsburgh, pp 144–152
Chen WF, Liu XL (1990) Limit analysis in soil mechanics. Amsterdam, Elsevier
Chen WF, Giger MW, Fang HY (1969) On the limit analysis of stability of slopes. Soils Found 9(4):23–32
Cortes C, Vapnik VN(1995) Support vector networks. Mach Learn 20:273–297
Cristianini N, Shawe-Taylor J (2000) An introduction to support vector machine. University Press, London, Cambridge
Dibike YB, Velickov S, Solomatine D, Abbot MB (2001) Model induction with support vector machine: introduction and application. J Comput Civil Eng 15(3):208–216
Fellenius W (1936) Calculation of stability of earth dams. In: Transactions 2nd Congress on large dams, Washington 4:445
Fletcher R (1987) Practical methods of optimization. Wiley, Chichester, Newyork
Gualtieri JA, Chettri SR, Cromp RF, Johnson LF(1999) Support vector machine classifiers as applied to AVIRIS data. In: The summaries of the 8th JPL airbrone earth science workshop
Hoek E, Bray JW (1981) Rock slope engineering, 3rd edn. Institution of Mining and Metallurgy, London
Hudson JA (1992) Rock engineering—theory and practice. Ellis Horwood, West Sussex
Karal K (1977a) Application of energy method. J Geotech Eng Div ASCE 103(5):381–399
Karal K (1977b) Energy method for soil stability analyses. J Geotech Eng 103(5):431–447
Kecman V (2001) Learning and soft computing: support vector machines, neural networks, and fuzzy logic models. The MIT Press, Cambridge
Khan MS, Coulibaly P (2006) Application of support vector machine in lake water level prediction. J Hydrol Eng 11(3):199–205
Lin PS, Lin MH, Lee TM (1988) An investigation on the failure of a building constructed on hillslope. In: Bonnard (ed) Landslides. Balkema, Rotterdam 1:445–449
Madzie E (1988) Stability of unstable final slope in deep open iron mine. In: Bonnard (ed) Landslides. Balkema, Rotterdam 1:455–458
MathWork Inc (1999) Matlab user's manual, Version 5.3. The MathWorks, Inc, Natick
Michalowski RL (1994) Limit analysis of slopes subjected to pore pressure. In: Srirwardane, Zaman (eds) Proceedings of the conference on comp. methods and advances in geomech. Balkema, Rotterdam
Michalowski RL (1995) Slope stability analysis: a kinematical approach. Geotechnique 45(2):283–293
Michalowski RL (2002) Stability charts for uniform slopes. J Geotech Geoenviron Eng ASCE 128(4):351–355
Morgenstern NR, Price VE (1965) The analysis of the stability of general slip surfaces. Geotechnique 15(1):79–93

Mukherjee S, Osuna E, Girosi F (1997) Nonlinear prediction of chaotic time series using support vector machines. In: Proc. IEEE workshop on neural networks for signal processing, vol 7. Institute of Electrical and Electronics Engineers, New York, pp 511–519

Muller KR, Smola A, Ratsch G, Scholkopf B, Kohlmorgen J, Vapnik VN (1997) Predicting time series with support vector machines. In: Proc. int. conf. on artificial neural networks. Springer, Berlin, pp 999

Osuna E, Freund R, Girosi F (1997) An improved training algorithm for support vector machines. In: Proc. IEEE workshop on neural networks for signal processing, vol 7. Institute of Electrical and Electronics Engineers, New York, pp 276–285

Park D, Rilett LR (1999) Forecasting freeway link ravel times with a multi-layer feed forward neural network. Comput Aided Civil Infrastruct Eng 4:358–367

Sakellatiou MG, Ferentinou MD (2005) A study of slope stability prediction using neural networks. Int J Geotech Geol Eng 23:419–445

Scholkopf B (1997) Support vector learning. R. Oldenbourg, Munich

Sincero AP (2003) Predicting mixing power using artificial neural network. EWRI World Water and Environmental

Smola AJ (1996) Regression estimation with support vector learning machines. Master's Thesis: Technische Universitat Munchen, Munchen, Germany

Smola AJ, Scholkopf B (2004) A tutorial on support vector regression. Stat Comput 14:199–222

Vapnik VN (1995) The nature of statistical learning theory. Springer, New York

Vapnik VN, Golowich S, Smola A (1997) Support method for function approximation regression estimation and signal processing. In: Mozer M, Petsch T (eds) advance in neural information processing system, vol 9. The MIT press, Cambridge

Vapnik VN (1998) Statistical learning theory. Wiley, New York