S. Saitta, P. Kripakaran, B. Raphael and I. F. C. Smith. Feature Selection using Stochastic Search: An Application to System Identification, in Journal of Computing in Civil Engineering, vol. 24, num. 1, p. 3-10, 2010. http://cedb.asce.org

Feature Selection using Stochastic Search: An Application to System Identification

Sandro Saitta[1], Prakash Kripakaran[2], Benny Raphael[3] and Ian F.C. Smith[4]

Abstract

System identification using multiple-model strategies may involve thousands of models with several parameters. However, only a few models are close to the correct model. A key task involves finding which parameters are important for explaining candidate models. The application of feature selection to system identification is studied in this paper. A new feature selection algorithm is proposed. It is based on the wrapper approach and combines two algorithms. The search is performed using stochastic sampling and the classification uses a support vector machine strategy. This approach is found to be better than GA-based strategies for feature selection on several benchmark data sets. Applied to system identification, the algorithm supports subsequent decision making.

Keywords: feature selection, wrapper, support vector machine, global search, system identification, decision support.

[1] Former grad. student, Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 18, 1015 Lausanne, Switzerland

[2] Postdoctoral researcher, Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 18, 1015 Lausanne, Switzerland

[3] Assistant Professor, School of Design and Environment, National University of Singapore (NUS), 4 Architecture Drive, Singapore 117566

[4] Professor, Ecole Polytechnique Fédérale de Lausanne (EPFL), Station 18, 1015 Lausanne, Switzerland, Ian.Smith@epfl.ch

Introduction

More and more structures are equipped with measurement systems in order to determine real in-service behavior. However, interpreting the data produced by these measurements is not easy. During the last decade, many researchers have applied system identification and model updating techniques for making structural health assessments using measurements (Catbas et al. 2007; Farrar and Jauregui 1998). System identification (Ljung 1999) is a model-based reasoning approach that involves determining the state of a system and values of system parameters from observed responses. While model-free methods have also been proposed using signal analysis methods, for example (Posenato et al. 2008), models provide more support for structural management tasks.

Traditionally, system identification is treated as an optimization problem to determine the values for model parameters such that the difference between model predictions and measurements is minimized. This approach is not reliable because different types of modeling and measurement errors are present (Catbas et al. 2007; Sanayei et al. 1997). Moreover, they may compensate each other such that the global minimum indicates models that are far from the model representing the correct state of the system (Robert-Nicoud et al. 2005b). Therefore, instead of optimizing one model, Robert-Nicoud et al. (2005b) proposed the selection of a set of candidate models such that their prediction errors lie below a certain threshold value. In this work, a model is defined as a distinct set of values for a set of parameters. Modeling assumptions define the parameters for the identification problem. The set of model parameters may consist of quantities such as elastic modulus, connection stiffness and moment of inertia. The threshold is computed

using an estimate of the upper bound of errors due to modeling assumptions as well as measurements (Saitta et al. 2008).

Complex structures often involve many model parameters leading to a large multi-dimensional space of candidate models. The set of candidate models is iteratively filtered using results from measurement-interpretation cycles for system identification. Since all candidate models are equally capable of representing the structure, identification of a single model is often not possible without advanced methods for interpretation and filtering. Furthermore, visualization of spaces of candidate models is difficult without computing support. Earlier research by the authors investigated data mining techniques such as decision trees, k-means clustering and principal component analyses to estimate the number of model classes and to visualize more effectively the model space (Saitta et al. 2005).

In addition to knowing the number of model classes, engineers may also benefit from knowledge of the most relevant parameters with respect to the set of candidate models. Values of a small subset of model parameters may determine whether or not a given model is a candidate model. Such knowledge reduces the dimensionality of the model space and supports subsequent decision-making. For example, the stiffness of a connection at a particular location may be an important feature of a set of candidate behavior models of a truss bridge. This implies that the parameter has a strong correlation with the measured response. Changes in measured values may indicate that this connection stiffness has changed due to damage or deterioration. Engineers use such information for inspection and repair.

Key parameters of a data set can be found using a concept known as feature selection (Dash and Liu 1997). Feature selection techniques may employ a wrapper-based approach where a classification algorithm is used to find the best set of features. These methods are popular since it offers the flexibility of using any search strategy (Kohavi and John 1998). Wrapper-based

approaches that combine support vector machines (SVM) - a classifier technique with search methods such as simulated annealing (SA) and genetic algorithms (GA) have already been proposed. Probabilistic Global Search Lausanne (PGSL) is a search technique that has shown superior performance when compared with GA and SA for continuous variables (Raphael and Smith, 2003a). Since the performance of a wrapper-based approach is dependent on the search method used, combining PGSL with SVM may result in a better feature selection method.

This paper presents a new wrapper-based feature-selection algorithm that offers better performance than existing feature selection methods. It combines support vector machine (SVM) and a global search algorithm (PGSL). The next section provides an overview of feature selection techniques. This is followed by an introduction to PGSL and SVM algorithms. The new algorithm is tested on benchmark data sets in order to compare its performance with existing techniques. The algorithm is then applied to the system identification task and the case of the Schwandbach Bridge in Switzerland illustrates the approach.

Feature selection techniques

Feature selection (Dash and Liu 1997) is a method used to reduce the number of features (parameters) before applying data mining algorithms. Irrelevant features may have negative effects on prediction tasks. Moreover, the computational complexity of a classification algorithm may suffer from the excessive dimensionality caused by several features, often referred to as the *curse of dimensionality*. When a data set has too many irrelevant variables and only a few examples, over fitting is likely to occur. From an engineering point of view, data are best characterized using as few variables as possible (Cheng et al. 2007).

Feature selection techniques can be classified into three main categories (Tan et al. 2006): embedded approaches (feature selection is a part of the classification algorithm), filter approaches (features are selected before the classification algorithm is used) and wrapper approaches (the

classification algorithm is used to find the best subset of attributes). Due to their definition, embedded approaches are limited since they only suit a particular classification algorithm. As noted in Molina et al. (2002), a relevant feature is not necessarily relevant for a given classification algorithm. Filter methods, however, make the assumption that the feature selection process is independent of the classification step. The work done by (Kohavi and Sommerfield 1995) recommends replacement of the filter approach by wrappers. This usually provides better results, at the expense of more computation (Weston et al. 2001). The over fitting problem is avoided by using a *k*-fold cross-validation strategy (Hsu et al. 2003). The accuracy of the classification algorithm may be used as the objective function of the search strategy.

Several feature selection methods exist in the literature. A comprehensive study of feature selection techniques is given in Saitta (2008). Individual ranking procedures are often called naive methods. The idea is to individually rank each feature at a time, according to its prediction power. This technique is valid only if every feature is independent, which is usually not the case in practice. Caruana and Freitag (1994) examine five hill-climbing procedures for feature selection. The main limitation of these methods is that they are greedy strategies. Greedy strategies are a class of search methods that choose only the best solution for exploration in subsequent iterations. Inferior solutions that may eventually lead to the global optima are ignored. Thus, greedy strategies are susceptible to local optima. To avoid being stuck in local optima of the feature selection objective function, random-based search strategies are used instead of greedy-like strategies. Using random-based search strategies is appropriate since the feature selection problem is exponential (Oh et al. 2004). An advantage of random-based search strategy is the avoidance of the monotonic assumption made by sequential methods (Yang and Honavar 1998). Lin et al. (2006) propose to combine simulated annealing (SA) with support vector machine (SVM) for feature selection and hyper-parameter optimization. Several studies have also been carried out using genetic algorithms (GA) for feature selection (Huang and Liu 2006; Yang

and Honavar 1998). Hybrid GA procedures have been proposed as well (Huang et al. 2007; Oh et al. 2004). Both SA and GA have a wide range of hyper-parameters to tune before obtaining convincing results (Kudo and Sklansky 2000; Oh et al. 2004). For SA, they are annealing schedule, number of loops, initial temperature and transition rate. For GA, they are population size, crossover rate, mutation rate and number of generations. If tuning is not carried out correctly, this leads to poor results. PGSL is a stochastic search algorithm that has many advantages over GA and SA. In this paper, the performance of a feature selection technique combining PGSL with SVM is studied.

Probabilistic Global Search Lausanne (PGSL)

The aim of feature selection is to find a subset of *m* features from a total of *d* that best satisfy a given criterion. For a given subset, a feature is either present or not. When finding all possible subsets *m* among *d*, Equation (1) gives the number of possibilities:

$$\sum_{m=0}^{d} C_d^m = \sum_{m=0}^{d} \frac{d!}{(d-m)!m!} = 2^d \tag{1}$$

Therefore, according to Equation (1), the number of possible feature combinations is combinatorial. A methodology for treating combinatorial problems involves the use of stochastic search.

PGSL is a direct search algorithm that employs stochastic sampling to find the global minimum of a user defined objective function. PGSL has been successfully applied to optimization problems involving non-linear objective functions containing a large number of local minima (Raphael and Smith 2003a). It has proven its efficiency for structural control (Domer et al. 2003), system identification (Robert-Nicoud et al. 2005a) and leak detection (Raphael and Smith 2005).

PGSL has advantages over SA and GA regarding hyper-parameter tuning (Domer et al. 2003). It has only three parameters that can be fixed using a simple guideline proposed in Raphael and Smith (2003a). Moreover, PGSL gives competitive results when compared to SA and GA (Domer et al. 2003; Raphael and Smith 2003a; Raphael and Smith 2005). PGSL performs global search through sampling the solution space using a probability density function (PDF). PGSL has three tuning parameters (for more details, see Raphael and Smith (2003a)):

- *NS*: number of samples (sampling cycle)
- *NFC*: number of loops in the focusing cycle
- *NSDC*: number of loops in the sub-domain cycle

At the beginning of search, a uniform PDF is assumed for the entire search space so that solutions are generated randomly. When good solutions are found, probabilities in those regions are increased so that more intense sampling is carried out in regions containing good solutions. The key assumption is that better sets of solutions are found in the neighborhood of good sets of solutions. The search space is gradually reduced so that convergence is achieved. The total number of PGSL iterations corresponds to the product of these three tuning parameters (*NS*·*NFC*·*NSDC*) or the satisfaction of a convergence criterion.

Support Vector Machine (SVM)

SVM have been successfully applied in domains such as text classification (Zhuang et al. 2005) and face recognition (Kotsia and Pitas 2007) among others. SVM hyper-parameters can be found through grid search (Soares et al. 2004). Chapelle et al. (2002) propose a gradient descent algorithm. In Luxburg et al. (2004), data compression is used on the training labels for hyper-parameter selection. Although a hybrid Monte Carlo technique is proposed in Gold and Sollich (2003), it is computationally expensive. As proposed in Frohlich et al. (2003) for GA, in the

approach suggested in this thesis, selection of SVM hyper-parameters is done throughout the feature selection process using PGSL. It has also been observed that SVM can suffer from irrelevant features (Rakotomamonjy 2003; Weston et al. 2001).

SVM are based on two concepts: the kernel trick and a separating hyperplane. The kernel is a function that transforms non-linear relationships from the initial space into linear relationships in order to discover relationships more easily in the feature space. A kernel ***K(x,y)*** is a function that evaluates the inner product between data points in some space:

$$K(\mathbf{x}, \mathbf{y}) = \varphi(\mathbf{x}) \cdot \varphi(\mathbf{y}) \tag{2}$$

where φ is an unknown mapping function. SVM is a margin classifier that can benefit from the kernel trick. A test instance **z** is classified using the decision function (separating hyperplane) of the non-linear SVM given below:

$$y = sign\left(\sum_{i=1}^{n} y_i \lambda_i K(\mathbf{x_i}, \mathbf{z}) + b \right) \tag{3}$$

where *n* is the number of training samples, $y_i \in [1,2]$ is the class label of the training example (for binary classification) x_i, $\lambda \in \lambda_1, \ldots, \lambda_n$ are the Lagrange multipliers, $K(\mathbf{x_i},\mathbf{z})$ is the chosen kernel function and *b* is a parameter related to the decision boundary. Training a SVM is done by minimizing the following objective function:

$$L(\lambda) = \sum_{i=1}^{n} \lambda_i - \frac{1}{2} \sum_{i,j} \lambda_i \lambda_j y_i y_j K(\mathbf{x_i}, \mathbf{x_j}) \tag{4}$$

subject to the following constraints:

$$\sum_{i=1}^{n} \lambda_i y_i = 0 \tag{5}$$

$$0 \le \lambda_i \le C, \forall i \tag{6}$$

where *C* is a SVM tuning parameter representing the penalty for misclassifying training examples. The SVM formulation described here is for binary classification problems. Methods are available for multi-class SVM, for example in Weston and Watkins (1998). The choice of the kernel $K(x_i,x_j)$ is important and generally depends on the application domain. The most commonly used kernel function is the Gaussian:

$$K(\mathbf{x},\mathbf{y}) = e^{-\frac{\|\mathbf{x}-\mathbf{y}\|^2}{2\sigma^2}} \tag{7}$$

The main reason for using a Gaussian kernel is that it has only one parameter (standard deviation, σ) to tune. Furthermore, it has provided good results in several applications. Other types of kernel have been examined in the literature and new kernels can be created (Cristianini and Shawe-Taylor 2000).

PGSL and SVM for Feature Selection

A wrapper approach is characterized by four aspects (Kohavi and John 1998). In the proposed algorithm, they are as follows:

- A state space of size: 2^d, where *d* is the total number of features
- An initial state: initial seed in PGSL
- A termination condition: PGSL maximum number of iterations
- A search algorithm: PGSL

The PGSL-SVM methodology combines global search (PGSL) with support vector machine (SVM). The strategy is founded on the proposition that feature selection and classification stages should be optimized together and not separately. Figure 1 shows the flowchart for the overall wrapper feature selection procedure.

First, one third of the data set is randomly taken to be the testing set (*Randomly divide data set* step). To avoid any over-fitting bias within results (Reunanen 2003), this test set is only used once, at the end of the process (*Evaluation of accuracy* step). PGSL is started with a random initial vector of dimension $d+2$ (*Feature selection* step). The first d values are rounded to either 1 or 0 (since PGSL uses continuous variables), respectively representing selected and non-selected features. The last two values are tuning parameters C and σ for Gaussian kernels in SVM.

The objective function that is minimized by PGSL is the classification error rate of the SVM. In the *Objective function (SVM)* step, a SVM with 10-fold cross-validation is run. The mean value of the 10 obtained error rates is given back to PGSL as the value of the objective function to minimize. If the total number of PGSL iterations is less than the limit, the loop continues. Otherwise, the feature subset corresponding to the minimum error is returned by PGSL. These features are selected from both the training and test sets to respectively train and evaluate the final accuracy of the SVM (*Evaluation of accuracy* step). The result of this overall procedure (Figure 1) is averaged over five separate runs.

The generalization accuracy is not the only relevant criterion for evaluating a feature selection strategy. Other factors are also of importance. The number of calls to the objective function is crucial for comparison, since estimating the generalization error using 10-fold cross-validation is expensive in terms of computational time (for each PGSL iteration, 10 SVM are trained). This is a good estimator of the computational complexity of the feature selection process. Since it is not related to a specific computer, the values for different wrapper approaches are easily comparable. Therefore, while the accuracy and the number of selected features are observed, the number of calls to SVM is fixed to be nearly the same (see Table 1). The number of features selected is also important. The fewer the number of features, the smaller is the amount of memory/time needed for the classification algorithm. In addition, a small number of features helps in understanding the data.

Due to the non-deterministic nature of PGSL and the size of the feature subset space (2^d), the approach is not guaranteed to find the best feature subset (the one that gives the best SVM results). However, the aim is to find a good subset of features that closely resembles the performance of the best solution. Therefore, the stability of the feature selection process is not studied. A feature selection algorithm that is stable is more *deterministic* than another one, however it does not mean that it performs better or it has found the best solution. The proposed approach is compared with random and GA-based feature selection to show its efficiency.

Benchmark tests

The PGSL-SVM approach is tested on several data sets from the University of California Irvine (UCI) database (Merz and Murphy 1996) and later, on real examples in system identification. The purpose of testing the approach on UCI data sets is to compare the performance of the proposed strategy with existing methods. As solutions to these data sets are available, they serve as benchmark tests for feature selection approaches. Obtaining good performance on these sample data sets is necessary before applying the approach to more complex civil engineering tasks such as structural system identification. Data sets have been chosen on the basis of their low number of missing values and their numeric features. Entries containing missing values have been discarded in the data preparation step to avoid issues related to missing data. Data sets are standardized with a zero mean and unit standard deviation. Feature selection has been performed using PGSL and SVM codes, both called with MATLAB.

Results

In this experiment, PGSL tuning parameters are set according to the instructions in the original paper by Raphael and Smith (2003a) and after experimental testing. Values are fixed as follows:

NS=2, *NFC=2·d* and *NSDC=2*, where *d* is the total number of features. SVM tuning parameters are fixed using PGSL ($C \in [0,100]$, $\sigma \in [0,10]$).

Cross-validation strategies are subject to over-fitting (Kohavi and Sommerfield 1995; Reunanen 2003). Due to the high number of possible feature subsets, a feature subset may be found that is better than others on only these particular cross-validation folds. Therefore, an additional subset that has never been used for the feature selection process is used as the test set (*Evaluation of accuracy* step).

Often, tuning parameters of SVM are manually set to a particular value manually as in Mao (2004). However, in this study, tuning parameters are set using a strategy depending on the method used. Four different methods are compared:

- **SVM**: Support vector machine without feature selection. SVM tuning parameters are chosen through a grid search ($C \in (1,10,100)$, $\sigma \in (0.1,1,10)$).
- **RAND-SVM**: Random selection of parameters. SVM with random feature selection. SVM tuning parameters are fixed to be the same as for SVM (see above).
- **GA-SVM**: GA feature selection combined with SVM. GA tuning parameters are based on the work by Yang and Honavar (1998). Probabilities of crossover and mutation are 0.6 and 0.001 respectively. Population size and number of generations are fixed, for each data set, so that the total number of GA evaluations is the closest to PGSL.
- **PGSL-SVM**: PGSL feature selection combined with SVM. SVM tuning parameters are modeled as PGSL parameters ($C \in [0,100]$, $\sigma \in [0,10]$). The total number of evaluations is dependent on the total number of features in the data set.

For large data sets (more than 200 samples), one third of the data is used as the test set (*Evaluation of accuracy* step). For data sets with less than 200 samples, no separate test set is

used. In the latter case, the *Evaluation of accuracy* step is done with a 10-fold cross-validation. Results of the four methods are given in Tables 1 and 2. Numbers within brackets in the first column of Table 1 indicate the dimensionality of data sets. For each strategy, 5 independent runs are made. Mean and Std in Table 1 represent the statistical mean and standard deviation over the 5 runs. Since RAND-SVM randomly chooses the number of selected features within each run, the table does not give this information. Table 2 provides the number of calls to the SVM 10-fold cross validation procedure. The second column in Table 2 is the number of samples for each data set.

An improvement in results when using GA-based or PGSL-based feature selection over standard SVM is visible for most data sets. *WDBC*, *Cancer wisconsin* and *Hungarian* show similar results when using either feature selection or standard SVM. This is due to their small initial number of features and their importance for classification. Only with *Cleveland* is the feature selection clearly performing worse. This may be due to the fact that every feature is important in explaining the different classes. Therefore, deleting even one of them significantly reduces classification accuracy.

Valuable improvements in classification accuracy are observed on several data sets. On the *Ionosphere* data set, GA-SVM and PGSL-SVM are better than SVM by 10.2% and 8.2% respectively. On *Zoo*, improvements are of 15.9% and 19.9%. Finally, the best improvements are shown for the *Hepatitis* data set, with accuracy increases of 36.8% and 38.9%.

Regarding GA-SVM and PGSL-SVM, it is noted that their classification accuracy is nearly the same on average. PGSL-SVM performs marginally better on 6 data sets out of 11. This indicates that both strategies are equivalent in their generalization ability. A more interesting result is the mean number of features selected. For 8 data sets out of 11, PGSL-SVM finds sets with less number of features than GA-SVM, for the same order of accuracy. This is due to the fact that

SVM tuning parameters are better fixed through PGSL-SVM. On *WDBC*, GA-SVM and PGSL-SVM find respectively 16 and 13 features for a difference of 1% in classification accuracy. PGSL-SVM has thus an improvement of 19.8% in the number of features. On *Lung cancer* and *Sonar*, improvements are 10.9% and 19.3% respectively.

The PGSL-SVM feature selection has two advantages over GA-SVM. First, with GA, the SVM tuning parameters have to be coded to match the usual binary format. This is not needed in the case of PGSL which uses continuous values. Second, PGSL has less tuning parameters to fix than GA. While GA has at least four tuning parameters, PGSL has a simple guideline concerning three variables. The main limitation of the proposed methodology, as with every wrapper-based approach, is the time consuming process of the classification algorithm evaluation. This time is further increased with standard cross-validation strategies.

The speed of convergence of GA and PGSL is dependent on the way their tuning parameters are fixed. Convergence studies on PGSL have been carried out in Raphael and Smith (2003b). In order to carry out a fair comparison between GA and PGSL search strategies, it is ensured that the number of calls to SVM is similar. Details of the number of calls to the 10-fold cross-validation procedure using SVM is given in Table 2. PGSL and GA are stopped when their respective numbers of iterations are achieved.

To summarize, a new feature selection algorithm using global search and SVM in a wrapper approach has been proposed. Experiments on several data sets have led to the following conclusions.

- PGSL-SVM is an efficient feature selection strategy. It performs as well as GA-SVM for feature selection on various data sets. Also, the PGSL-SVM finds subsets with a smaller number features than GA-SVM for the same order of accuracy and time.

- PGSL uses continuous values and this helps find the optimal tuning parameters of SVM during the feature selection process.
- PGSL is easier to use since it has less tuning parameters than GA-based strategy. This is important since bad tuning can lead to poor results. Furthermore, correct tuning is often time consuming.

Feature Selection in System Identification

Schwandbach bridge

To illustrate the feature selection algorithm for system identification, the Schwandbach Bridge (designed by Maillart in 1933) is taken as a case study (Figure 2). This structure is inspected periodically and has been the subject of many verifications as codes have improved, for example Salvo (2006). The Schwandbach Bridge is now a pedestrian bridge, although it could be reopened for traffic. Deflection measurements have not been carried out since the 1930s and while the bridge shows no visible evidence of deterioration, the question of taking measurements arises periodically. In Switzerland, bridges are traditionally measured for changes in deflection at mid-span during load tests. A single model (usually the design model) is used with the deflection measurement and the loading to determine values for parameters that have some uncertainty, such as the elastic modulus multiplied by the moment of inertia, E·I. However, this bridge is too complex for such rudimentary model-calibration strategies.

While many assumptions are acceptable at the design stage for achieving safety and serviceability, they are not appropriate for interpreting measurements. For example, there is no physical hinge at the extremities of the vertical spandrel elements. These connections cannot be assumed to be fixed either since even small amounts of cracking reduce connection stiffness. Furthermore, not all connections are expected to have the same stiffness due to factors such as

relative slenderness and varying locations on the structure. The Schwandbach Bridge has 20 such connections. They are shown in Figure 3 using open circles. The methodology is used to select relevant model parameters (values for connection stiffness) that can explain bridge behavior.

The number of permutations and combinations of modeling assumptions – values for connection stiffness - results in several tens of thousands of possible models. Although the Schwandbach bridge has important technical and historical attributes, these conclusions are equally valid for most ordinary structures of moderate complexity.

Bridges are often tested periodically using static loads to check for strength degradation. The response of the bridge for trucks positioned on the bridge is measured using sensors. Engineers estimate the stiffness of the bridge from measured responses and compare those with results from previous tests. Such a scenario is simulated for the Schwandbach Bridge. Measured responses are used to find the stiffnesses of the connections at the extremities of the vertical spandrel elements. Thus the stiffnesses of the connections are the model parameters. Loads equivalent to that of two trucks on the bridge are simulated. Sensors are assumed at 5 locations on the structure given by positions 1, 6, 10, 13 and 18 (see Figure 3).

Stochastic search is used to find a set of 1000 models as described in Robert-Nicoud et al. (2005b). Among these models, 500 are candidate models. Candidate models are those for which the difference between measurements and predictions is below a certain threshold value set as 8 μrad They correspond to the models that closely represent the structure behavior (in this case, the Schwandbach bridge).The other 500 models in the set are not candidate models. More details on this case study can be found in Saitta et al. (2008).

Results

The case study introduced in the previous section is used for illustrating feature selection. The starting point is thus a matrix of 1000 rows and 21 columns. The number of rows corresponds to the number of models. The first 20 columns contain, for each model, the value for a parameter, i.e., the stiffness value of a connection. The last column corresponds to the class label. A candidate model is labeled with 1 and a non-candidate model with 2. At this point, the PGSL-SVM methodology is run 5 times. The size of the test set is fixed as one third of the data set (33%). Results obtained are given in Table 3.

First, it is observed that the standard deviation of the test accuracy is low. This means that results of the 5 different runs are close. Regarding the number of features, it is observed that around 11 connection stiffnesses are selected, in mean, out of 20. Thus, about half of the connection stiffnesses are useless in separating candidate from non-candidate models.

For this experiment, the number of PGSL iterations is set to 160 and 5 independent runs are averaged. The best test accuracy (97.9%) corresponds to the selection of the following parameters: p_2, p_4, p_7, p_9, p_{11}, p_{12}, p_{14}, p_{15}, p_{16}, p_{17}, p_{18}, p_{19} and p_{20}. Therefore, with these 13 connection stiffnesses, one can argue that a model is candidate with more than 97.9% accuracy. These 13 parameters are shown with black dots in Figure 4.

This set of 13 features is of importance for engineers. It can be used to support further decisions. For example, the other 7 connection stiffnesses are not important in identifying candidate models. Variations at these positions do not help engineers for the system identification task. Since these 13 features have been selected, it means that the other 7 either contain similar information (they are redundant) or no information at all. The 13 connection stiffnesses are independent from each other since they contain no or few redundant information. This may change assumptions of engineers about the structure. Therefore, feature selection can give useful

information to engineers who must decide on subsequent sensor placement and evaluate the validity of modeling assumptions.

Conclusions

A new feature selection algorithm based on the wrapper concept is proposed in this paper. Feature selection is found to be helpful for interpreting system identification results, especially when the number of candidate models is large. The most important parameters of candidate models are identified and redundant parameters are eliminated. When engineers are given model parameters that best separate candidate from non-candidate models, they can better understand why some models become candidates. The advantage of such knowledge is that it is easily readable by engineers. Feature selection can reduce dimensionality of the problem. Therefore, it gives engineers a better understanding of the candidate model space. By knowing the important features of a measured structure, engineers can take better decisions with respect to structural management. The following more specific conclusions come out of this paper:

- The newly developed algorithm, PGSL-SVM, is an efficient feature selection strategy. It performs better than existing algorithms such as the GA-SVM for feature selection on various benchmark data sets. These tests provide confidence in applying the algorithm to practical civil engineering tasks.
- The PGSL-SVM strategy finds subsets with a smaller number of features than GA-SVM for the same order of accuracy and in the same amount of time. Fewer parameters are undoubtedly advantageous since unique system identification is faster than with the full parameter set.
- The strategy involving PGSL has less tuning parameters than the GA-based strategy. The number of tuning parameters is important since bad tuning can lead to poor results and the possibility of bad tuning increases with the number of tuning parameters.

- Feature selection supports system identification since it identifies parameters that are relevant for explaining candidate models. In the case study, stiffness values of 7 joints were found to have an insignificant influence on measurement data values.

Future work includes testing the methodology on measurements of dynamic excitation. Furthermore, the function to be minimized could be multi-objective. For example, in addition to minimizing the SVM error rate, one could also minimize the number of selected features.

Acknowledgements

This research was funded by the Swiss National Science Foundation, Grant No 200020-117670/1. The authors are grateful to Prof. E. Brühwiler, EPFL for providing details of the Schwandbach Bridge.

References

Caruana, R., and Freitag, D. (1994). "Greedy Attribute Selection." *International Conference on Machine Learning*, 28-36.

Catbas, F. N., Ciloglu, S. K., Hasancebi, O., Grimmelsman, K., and Aktan, A. E. (2007). "Limitations in Structural Identification of Large Constructed Structures." *Journal of Structural Engineering*, 133(8), 1051-1066.

Chapelle, O., Vapnik, V., Bousquet, O., and Mukherjee, S. (2002). "Choosing Multiple Parameters for Support Vector Machines." *Machine Learning*, 46(1-3), 131-159.

Cheng, H., Chen, H., Jiang, G., and Yoshihira, K. (2007). "Nonlinear Feature Selection by Relevance Feature Vector Machine." *Machine Learning and Data Mining in Pattern Recognition*, 144-159.

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

Dash, M., and Liu, H. (1997). "Feature selection for classification." *Intelligent Data Analysis*, 1(3), 131-156.

Domer, B., Raphael, B., Shea, K., and Smith, I. F. C. (2003). "A study of two stochastic search methods for structural control." *Journal of Computing in Civil Engineering*, 17(3), 132-141.

Farrar, C. R., and Jauregui, D. A. (1998). "Comparative study of damage identification algorithms applied to a bridge: I. Experiment." *Smart Materials & Structures*, 7(5), 704-719.

Frohlich, H., Chapelle, O., and Schölkopf, B. (2003). "Feature Selection for Support Vector Machines by Means of Genetic Algorithms." *Proceedings of the 15th IEEE International Conference on Tools with Artificial Intelligence*, 142-148.

Gold, C., and Sollich, P. (2003). "Model selection for support vector machine classification." *Neurocomputing*, 55(1-2), 221-249.

Hsu, C.-W., Chang, C.-C., and Lin, C.-J. (2003). "A Practical Guide to Support Vector Classification." National Taiwan University.

Huang, J., Cai, Y., and Xu, X. (2007). "A hybrid genetic algorithm for feature selection wrapper based on mutual information." *Pattern Recogn. Lett.*, 28(13), 1825-1844.

Huang, P.-W., and Liu, C.-L. (2006). "Using genetic algorithms for feature selection in predicting financial distresses with support vector machines." *IEEE International Conference on Systems, Man, and Cybernetics*.

Kohavi, R., and John, G. (1998). "The wrapper approach." Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 33-50.

Kohavi, R., and Sommerfield, D. (1995). "Feature Subset Selection Using the Wrapper Model: Overfitting and Dynamic Search Space Topology." *The First International Conference on Knowledge Discovery and Data Mining*, 192-197.

Kotsia, I., and Pitas, I. (2007). "Facial expression recognition in image sequences using geometric deformation features and support vector machines." *IEEE Transactions on Image Processing*, 16(1), 172-187.

Kudo, M., and Sklansky, J. (2000). "Comparison of algorithms that select features for pattern classiers." *Pattern Recognition*, 33, 25-41.

Lin, S.-W., Tseng, T.-Y., Chen, S.-C., and Huang, J.-F. (2006). "A SA-based feature selection and parameter optimization approach for support vector machine." *IEEE International Conference on Systems, Man, and Cybernetics*.

Ljung, L. (1999). *System Identification - Theory For the User*, Prentice Hall.

Luxburg, U., Bousquet, O., and Schölkopf, B. (2004). "A Compression Approach to Support Vector Model Selection." *Journal of Machine Learning Research*, 5, 293-323.

Mao, K. Z. (2004). "Feature Subset Selection for Support Vector Machines Through Discriminative Function Pruning Analysis." *IEEE Transactions on systems, man, and cybernetics*, 34(1), 60-67.

Merz, C. J., and Murphy, P. M. (1996). "UCI Machine Learning Repository." University of California, Irvine, School of Information and Computer Sciences, Irvine, CA.

Molina, L. C., Belanche, L., and Nebot, A. (2002). "Feature Selection Algorithms: A Survey and Experimental Evaluation." *Proceedings of the 2002 IEEE International Conference on Data Mining (ICDM'02)*.

Oh, I.-S., Lee, J.-S., and Moon, B.-R. (2004). "Hybrid genetic algorithms for feature selection." *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 26(11), 1424-1437.

Posenato, D., Lanata, F., Inaudi, D., and Smith, I. F. C. (2008). "Model-free data interpretation for continuous monitoring of complex structures." *Advanced Engineering Informatics*, 22(1), 135-144.

Rakotomamonjy, A. (2003). "Variable selection using SVM-based criteria." *Journal of Machine Learning Research*, 3, 1357-1370.

Raphael, B., and Smith, I. F. C. (2003a). "A direct stochastic algorithm for global search." *Journal of Applied Mathematics and Computation*, 146(2-3), 729-758.

Raphael, B., and Smith, I. F. C. (2003b). *Fundamentals of Computer-Aided Engineering*, Wiley.

Raphael, B., and Smith, I. F. C. (2005). "Engineering Applications of a Direct Search Algorithm, PGSL." *Proceedings of the 2005 ASCE Computing Conference*.

Reunanen, J. (2003). "Overfitting in Making Comparisons Between Variable Selection Methods." *Journal of Machine Learning Research*, 3, 1371-1382.

Robert-Nicoud, Y., Raphael, B., Burdet, O., and Smith, I. F. C. (2005a). "Model Identification of Bridges Using Measurement Data." *Computer-Aided Civil and Infrastructure Engineering*, 20(2), 118-131.

Robert-Nicoud, Y., Raphael, B., and Smith, I. F. C. (2005b). "System identification through model composition and stochastic search." *Journal of Computing in Civil Engineering*, 19(3), 239-247.

Saitta, S. (2008). "Data Mining Methodologies for Supporting Engineers during System Identification," Dissertation, EPFL, Lausanne.

Saitta, S., Kripakaran, P., Raphael, B., and Smith, I. F. C. (2008). "Improving System Identification using Clustering." *Journal of Computing in Civil Engineering*, 22(5), 292-302.

Saitta, S., Raphael, B., and Smith, I. F. C. (2005). "Data mining techniques for improving the reliability of system identification." *Advanced Engineering Informatics*, 19(4), 289-298.

Salvo, A. (2006). "Ponts de Robert Maillart." EPFL-MCS, Lausanne, Switzerland.

Sanayei, M., Imbaro, G., McClain, J. A. S., and Brown, L. C. (1997). "Structural Model Updating Using Experimental Static Measurements." *Journal of Structural Engineering*, 123(6), 792-798.

Soares, C., Brazdil, P., and Kuba, P. (2004). "A meta-learning method to select the kernel width in Support Vector Regression." *Machine Learning Journal*, 54(3), 195-209.

Tan, P.-N., Steinbach, M., and Kumar, V. (2006). *Introduction to Data Mining*, Addison Wesley.

Weston, J., Mukherjee, S., Chapelle, O., Pontil, M., Poggio, T., and Vapnik, V. (2001). "Feature selection for SVMs." *Advances in Neural Information Processing Systems*.

Weston, J., and Watkins, C. (1998). "Multi-class support vector machines." *CSD-TR-98-04*, Department of Computer Science, Royal Holloway, University of London.

Yang, J., and Honavar, V. (1998). "Feature subset selection using a genetic algorithm." Feature Selection for Knowledge Discovery and Data Mining, Kluwer Academic Publishers, 117-136.

Zhuang, D., Zhang, B., Yang, Q., Yan, J., Chen, Z., and Chen, Y. (2005). "Efficient text classification by weighted proximal SVM." *Fifth IEEE International Conference on Data Mining*, 538-545.

	SVM		RAND-SVM		GA-SVM				PGSL-SVM			
	Accuracy		Accuracy		Accuracy		# features		Accuracy		# features	
Data set	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**	**Mean**	**Std**
WDBC (30)	97.9	1.3	84.8	6.0	97.5	1.2	16.2	1.3	96.5	1.1	13.0	1.2
Cleveland (13)	86.5	2.5	57.6	2.7	84.1	2.5	8.4	0.9	80.8	2.6	7.0	2.1
Cancer wisconsin (9)	96.5	1.7	96.4	1.2	96.2	0.8	5.6	1.1	95.6	0.9	4.8	1.1
Ionosphere (34)	84.0	1.0	66.6	17.0	92.6	1.0	17.6	2.4	90.9	1.3	16.4	3.6
Wine (13)	93.5	0.5	92.3	3.3	98.2	0.5	8.2	1.3	98.7	0.4	7.8	1.5
Hepatitis (19)	56.5	0.6	63.5	3.1	77.3	3.2	8.6	1.1	78.5	2.7	6.4	1.7
Glass (9)	59.7	5.1	31.1	4.3	62.3	5.2	4.8	1.1	61.4	4.5	5.4	1.1
Hungarian (13)	81.2	1.7	70.3	7.6	78.1	3.1	5.8	0.8	79.8	4.9	6.4	0.6
Sonar (60)	77.1	7.2	43.5	22.6	81.5	4.3	30.0	2.5	83.2	7.9	24.2	2.5
Zoo (16)	81.4	1.5	88.2	6.3	94.3	2.0	7.8	1.5	97.6	0.6	8.8	0.8
Lung cancer (57)	73.2	5.9	77.3	5.5	88.3	3.4	25.6	3.1	89.2	5.2	22.8	4.8

Table 1: Comparison of accuracy for SVM, random feature selection (RAND-SVM), GA-based feature selection (GA-SVM) and PGSL-based feature selection (PGSL-SVM)

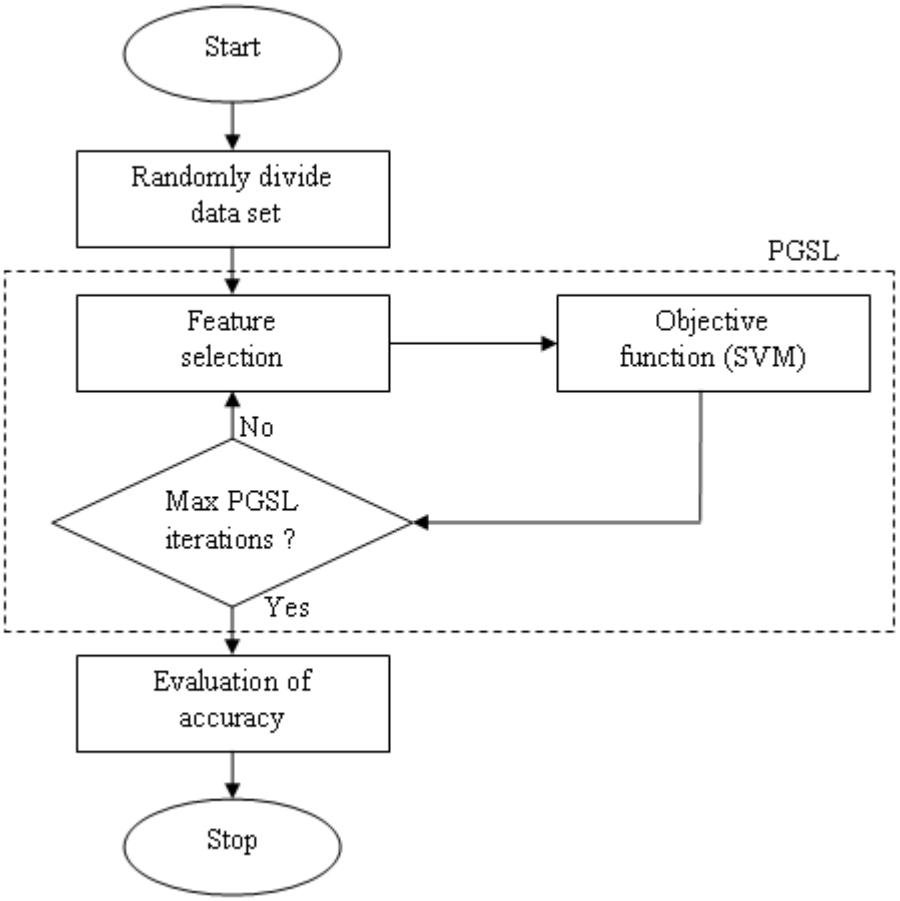
Data set	Size	GA-SVM	PGSL-SVM
WDBC	569	240	240
Cleveland	297	110	104
Cancer wisconsin	683	72	72
Ionosphere	351	272	272
Wine	178	110	104
Hepatitis	80	156	152
Glass	214	72	72
Hungarian	294	110	104
Sonar	208	506	480
Zoo	101	132	128
Lung cancer	27	462	448

Table 2: Number of GA and PGSL iterations during the runs in which each gave the best results

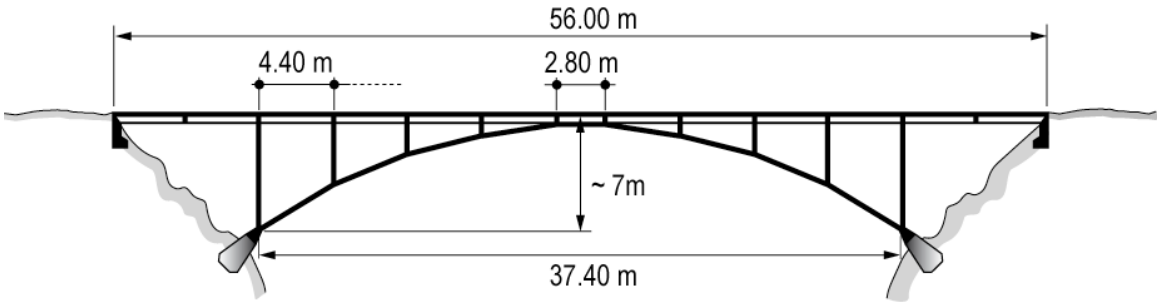
Information	Value
Mean test accuracy	97.2%
Standard deviation of test accuracy	0.6
Mean number of features	11.4
Standard deviation of number of features	2.1

Table 3: Results obtained for feature selection on 1000 models over 5 independent runs.

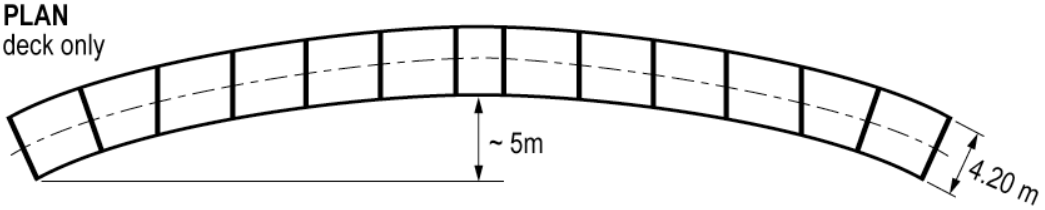
List of figures

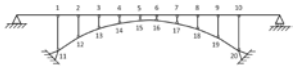

Start
Randomly divide data set
PGSL
Feature selection
Objective function (SVM)
No
Max PGSL iterations ?
Yes
Evaluation of accuracy
Stop

ELEVATION



PLAN
deck only




1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19
20