4 Enhanced Techniques for Analog Circuits Design Using SVM Models

Abstract. In order to improve the relatively slow convergence of GA, in the presence of large search spaces, and reduce the high consuming time of evaluation functions in analog circuit design applications, this chapter will discuss the use of learning algorithms. These algorithms explore the successive generation of solutions, learn the tendency of the best optimization variables and will use this knowledge to predict future values. In other words, these techniques employ data mining theory, used to manage large databases and huge amount of internet information, to discover complex relationships among various factors and extract meaningful knowledge to improve the efficiency and quality of decision making. In this chapter a new hybrid optimization algorithm is presented together with a design methodology, which increases the efficiency on the analog circuit design cycle. This new algorithm combines an enhanced GA kernel with an automatic learning machine based on SVM model (GA-SVM) which efficiently guides the selection operator of the GA algorithm avoiding time-consuming SPICE evaluations of non-promising solutions. The SVM model is here defined as a classification model used to predict the *feasibility region* in the presence of large, non-linear and constraints search spaces that characterize analog design problems. The SVM modeling attempts to constraint the search space in order to accelerate the search towards the feasible region ensuring a proper operation of the circuit.

4.1 Learning Algorithms Overview

Data mining consists of exploring data in order to discover unknown patterns and meaningful relationships in data, which may be used to make valid predictions. Within this technology data play an important role and the knowledge, extracted by the use of pattern recognition technologies as well statistical and mathematical techniques are the driven force in the new decision support systems. The adoption of this technology can increase the productivity in business or in the process where it was applied, since the same goals could be achieved or even improved with less investment in efforts and resources.

The technology behind data mining techniques is mostly based on inductive learning [1], where a model is constructed by generalizing from an adequate number of training samples collected from an historical database or coming from an experiment in which the sample is tested. Once built, the trained model can be applied to unseen examples to predict future trends and behaviors. This typical learning scenario is illustrated in Fig. 4.1 and it is known as a supervised learning approach. This differs from other approaches in what concerns to the feedback,

M.F.M. Barros et al.: Analog Circuits and Systems Optimization, SCI 294, pp. 89–107.
springerlink.com © Springer-Verlag Berlin Heidelberg 2010

received from its process during the learning stage. For example, in the reinforcement learning the feedback signal does not contain the knowledge of the environment, which is supplied to the learning machine in the supervised learning. Instead, the learning machine only receives a rating of its performance, often called reinforcement signal. In the unsupervised learning approach, the learning machine does not receive any feedback information at all, only the input samples. The learning machine is charged to reveal properties or knowledge hidden in the data, e.g. associating these data into groups or classes based on correlation of samples.

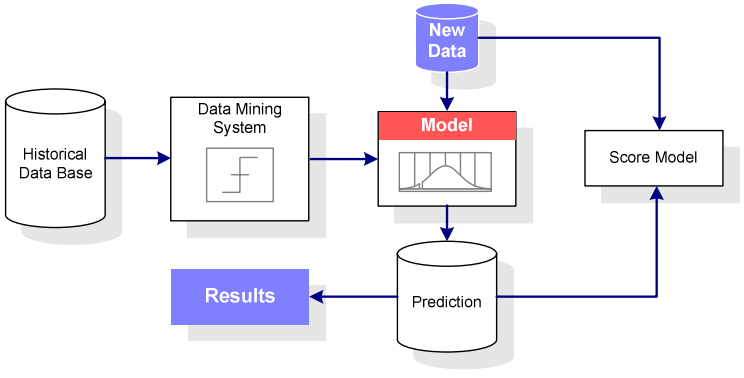


Fig. 4.1 Supervised learning approach

The data mining tools were originally developed to answer to specific problems in several areas of application and different knowledge domains, and so they inherited special characteristics that make each specific technique tailored to some type of problem. The most usual of these are:

- Classification and regression. These classes embrace the largest number of problems in the data mining domain [2]. In the classification problems, the learning machine creates a model to predict the class membership to which an entity belongs to, whereas in the regression case the model aims to predict a real-value variable based on the relationship between the other variables, assuming a linear or nonlinear relationship.
- Association and sequencing. Also known as market basket analysis, these techniques create models to discover hidden patterns of behavior, correlations among a set of objects generating an output in the form of descriptive rules, e.g., "75% of the customers who buy milk also buy bread and eggs". The sequencing technique is very similar to an association technique, but it includes description rules with information of time in the final analysis.
- Clustering. This technique seeks to identify a set of groups or clusters that defines the given data. Basically, it groups together entities or data points with similar behavior or properties, and creates different groups for dissimilar entities.

Advances in data mining were boosted by the progress in the fields of artificial intelligence (AI) and statistics. Fig. 4.2 provides a description of some of the most common data mining approaches used nowadays. Below, these techniques are briefly described.

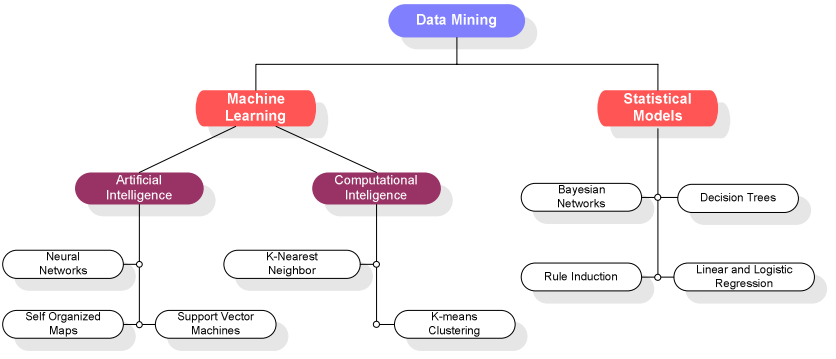


Fig. 4.2 Data mining technology

The regression technique implements a model based on observed data to forecast the output effect of a data item on the modeled system. In the simplest case, regression uses standard statistical techniques such as linear regression, which is modeled by a strait line that best fits the data and lately uses this line to predict values. The optimum model is obtained through the line that minimizes the sum of the square error from each data sample. The linear regression equation is described in Fig. 4.3.

However, for many real-world problems, predictions are very difficult to obtain because they may depend on complex interactions between multiple predictor

Linear regression model formulation:

$$Y = a + bX + e$$

where,

Y – dependent variable
X – independent variables
a – constant term
b – coefficient of indep. variables
e – the error term or residual

The "residual" e is a random variable with mean zero. The coefficients a and b are determined by the condition that the sum of the square residuals is as small as possible.

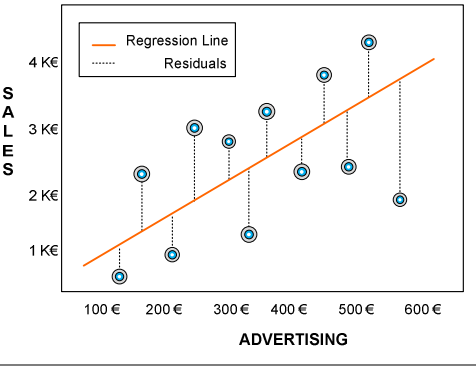


Fig. 4.3 Linear regression

variables. Therefore, more sophisticated algorithms are used for these cases such as, logistic regression, decision trees, neural nets and support vector machines.

Neural networks (NN) [3-8] are inspired on an early model of human brain function, whereas support vector machines (SVM) had their inspiration on the statistical learning theory [9]. Both have proven great efficiency either in classification as in regression type of problems. They require the configuration of model parameters in order to be efficient. The input for these models is limited to numerical data and the output is essentially predictive, i.e., they were designed to build models to forecast future behaviors but do not have mechanisms to summarize data and highlight their interesting properties. Due to this behavior, they are often referred to as "black box" technologies. Generally, the training of these models can be time consuming, although the predictions for new values are processed very fast. A great advantage of these algorithms is their ability to be used as an arbitrary function approximation constructed from past observations. This aspect is particularly useful in complex and expensive data analysis functions or even in situations where there is no defined function, but only a set of samples. Although both NN and SVM have common characteristics, they differ radically in one important aspect: SVM training always finds a global minimum [9].

The decision tree is a technique in which the resulting model is represented by graphic structure in a form of tree. One overview of this representation is illustrated in Fig. 4.4.

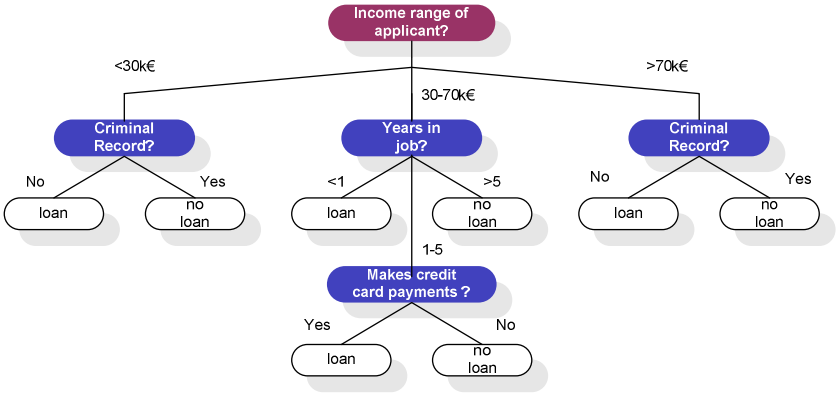


Fig. 4.4 Decision tree representation example

The tree representation helps to identify the important factors of the problem (the nodes) and how these factors have been affecting the outcomes of the decision in the past. The final decision is found in one of the leaf nodes at the bottom of the tree, after traveling from the root at the top and traversing several sub-nodes according to some test execution in each sub-node. The decision trees are mostly used for classification. The graphical representation is an attractive characteristic because it is easy to understand, which makes this technique become one of the most popular tools for data mining problems.

K-nearest neighbor (KNN) [10] is also a predictive technique suited for classification models. Unlike the other predictive techniques, it has no training phase once the training data represents simultaneously the model, thus models tend to be very large. The predictions for a new sample is done by looking for the group of similar characteristics and calculate the outcome value based on the most predominate class ("*k*" means the number of the nearest points with similar characteristics). The definition of this model is associated with a metric to measure the distances. The choice of metric is an important specification to take into account because the performance of the model depends on it.

K-means [10] is one of the simplest unsupervised learning algorithms tailored to solve the clustering problems. It is used to classify data, following a procedure that groups a given data set through a certain number of clusters (assume k clusters or subsets) defined by the user. The grouping routine minimizes the sum of squares of distances between data and the corresponding cluster centroid. When all samples have been assigned to a group which has the closest centroid, the algorithm recalculates the positions of the K centroids and repeats the process until the centroids get a stationary phase. Despite the simplicity, the k-means algorithm is also significantly sensitive to the initial randomly selected cluster centers and sometimes misses to find the most optimal configuration related to the global objective function minimization expressed in Fig. 4.5.

Objective function:

$$F = \sum_{j=1}^{k} \sum_{i=1}^{N} \left\| x_j^{(j)} - c_j \right\|^2$$

where,

$\left\| x_j^{(j)} - c_j \right\|$ is the chosen distance metric between a data point $x_j^{(j)}$ and the cluster centre c_j.

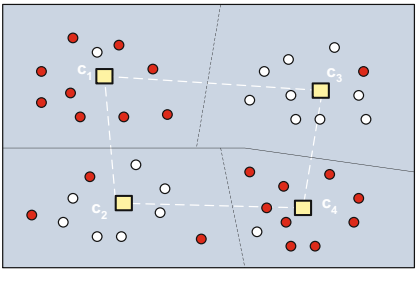


Fig. 4.5 K-means objective function

The next group of techniques has its origin in the *Naïve-Bayes* algorithm which uses the computation of probabilities as the main tool to make predictions. Naïve-Bayes is a classification technique that is not only predictive but also owns a descriptive feedback that describes the basic features and the interesting properties of the data. This approach assumes the statistically independence of all the independent variables which may not be true and is tailored to deal with categorical problems. The categorical limitation can be overcome to handle continuous data using bracket techniques that determine categories defined by limits of continuous data. Although technically simple to implement, the selection of the ranges can have a dramatic impact on the quality of the final model. The Naïve-Bayes concept is based on the relationship between dependent and independent variables and

produces conditional probabilities derived from observed frequencies in the training data. Extending the Bayesian technique to capture the interactions between pairs of non-independent columns is also possible, although the complexity and storage capacity will increase a lot. However, in its simple form (assuming independence of variables) Naïve-Bayes is considered an easy and time efficient exploratory tool.

The Table 4.1 briefly summarizes some of the major characteristics of the learning algorithms presented in Fig. 4.2.

Table 4.1 Classification of data mining techniques

Methods	*Easy of use & understand*	*Class*	*Problems*	*Notes*
Support Vector Machines (SVMs)	-	Supervised	Classification Regression	SVMs are considered one of the most effective machine learning tool with the ability to represent non-linear relationships and produce models that generalize well to unseen data. SVM training always finds a global minimum.
Artificial Neural Nets (ANNs)	-	Supervised	Classification Regression Clustering	Relation between weights and variables is difficult to interpret. Dificult to build the network structure. Require large amounts of time to train. Error decreases as a power of the training size.
Decision Trees (DTs)	++	Supervised	Classification Regression	Clear. A series of nested if/then rules. Relatively fast. Easy to translate into SQL queries
Nearest Neighbor Methods (e.g., kNN)	++	Supervised	Clustering Classification	It is fast and easy to use and understand. Ideal candidate for quickly building and testing classification models. Drawback: Models tend to be very large.
Splines (e.g. MARS: Multivariate Adaptive Regression S.)	+	Supervised	Regression	One of the most widely used statistical techniques for creating predictive models
Logistic Regression	+	Supervised	Regression	One of the most widely used statistical techniques for creating predictive models.
Rule Learning	+	Supervised Unsupervised	Classification	Understandable. The computation of probabilities of all combinations can be expensive!
K-means clustering	++	Unsupervised	Clustering Classification	Simplicity. Sometimes misses the most optimal configuration and is sensitive to the initial cluster centers.
Self organized maps (SOM)	-	Unsupervised	Classification	Similar to feed-forward neural net except that there is one output for every hidden layer node.
Bayesian networks	+	Supervised	Classification	Limits their inputs to categorical data.

4.1.1 SVM Classification Overview

SVMs belong to a class of supervised learning algorithms which are able to acquire knowledge from previous experiences and to apply the knowledge to predict future values [1],[9], [11-12]. This process is known as *memorization* and *generalization*. The modeling presented here is based on a supervising SVM approach to the two-class classification problem, where a set of training data of the form $S=\{(x_i; y_i),\ldots,(x_n; y_n)\}$ is observed, and the input $x_i \in X \subset R^d$ is a d-dimensional feature vector and the output $y_i \in \{+1,-1\}$ is the class label of x_i. The main goal is to train a discriminate function, which will be used to predict the labels for new inputs, minimizing the probability of classification errors.

Generally, the support vector classifier is implemented in a two step process. First, it is applied the kernel "*trick*", which provides a nonlinear mapping of the vectors x_i into a higher dimensional *feature space*. In the second step, a decision boundary hyperplane is created based on the *maximal-margin principle*. This process is illustrated in Fig. 4.6 where the input space of two classes originally inseparable, is mapped into a feature space, making it possible the separation of the two classes in a linear way.

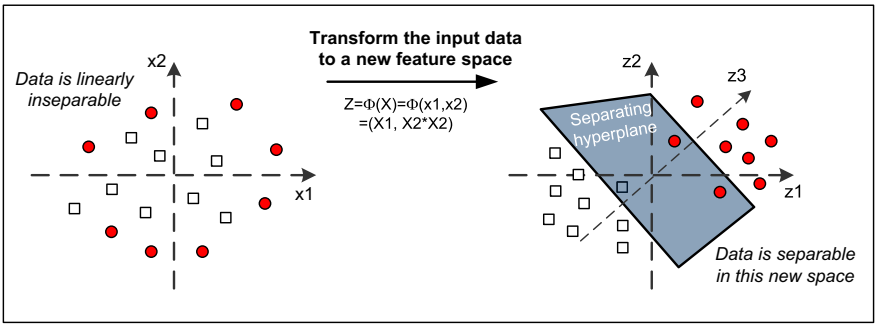


Fig. 4.6 Separating the data in a feature space

The SVM learning algorithm finds the *optimal separating hyperplane* (OSH) that maximizes the distance between the decision boundary between the two class groups and the closest point to the boundary, known as the margin, as illustrated in Fig. 4.7. The decision boundary points overlapping the margins are called *support vectors*. Support vectors are the most relevant in the decision process. The separating hyperplane in the feature space can correspond to a nonlinear decision boundary in the original input space. A more extended background on SVM concepts, issues and formulation, is presented in Appendix D.

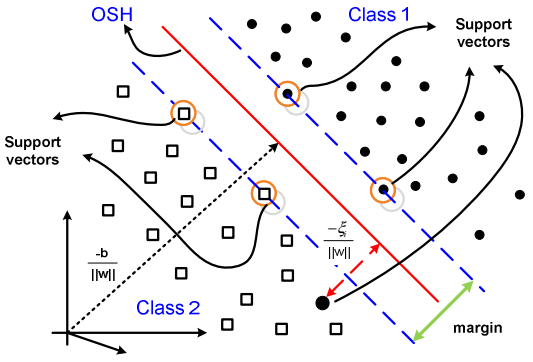


Fig. 4.7 Illustration of OSH hyperplane, margin and support vectors concept

4.2 GA-SVM Optimization Approach

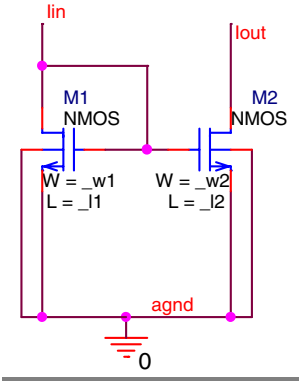
In this book the supervised learning algorithm belonging to the class of machine learning algorithm called SVM, was adopted to work together with the selected GA approach. The GA-SVM methodology explores the properties of *the sizing rules method*, commonly used in analog circuits, and produces a *feasibility* model of the functional space while the GA search engine is used to explore the *design space* and supply the SVM model with knowledge extracted from previous experiences. The SVM model is here defined as a classification model used to predict the *feasibility region*, in this context, the new SVM model will be referred as a *feasibility model*. Despite the strong theoretical foundations and recognized robust algorithm, the success of SVM implementations greatly depends on several intrinsic parameterization values and data preparation routines [13].

4.2.1 Feasibility Region Definition

One problem often found with numerical optimization methods is the generation of results considered *pathological*, that is, a result that on the one hand meets all specifications but on the other hand fails some basic design requirements (e.g. saturation of certain transistors) [14], leading to a malfunction circuit. This inconvenient behavior is derived from insufficient design specifications, where a circuit optimization problem is considered as a black box with a number of design parameter constraints and performances constraints. Expert IC designers learn how to deal with *pathological* sizing by manually constrain the circuit to ensure proper biasing and good behavior of performance metrics. For example, fixing all transistor lengths and applying device matching conditions is a common practice employed in analog circuit design. The methodology which attempts to automatically constrain a circuit in order to ensure proper operation is called the *sizing rules*

method [15]. Applying this methodology not only avoids the pathological designs but also improves the behavior of performance metrics and reduces sensitivity to operating conditions and process variations [16].

A generalized view of the sizing rules methodology points to the use of inequality constraints on electrical parameters (voltages and currents) in order to ensure the correct circuit operation. For example, [14],[17] introduces the concept of functional constraints and applies this concept to a simple CMOS current mirror. Functional constraints are a set of additional specifications defined analytically with a strong dependence on the application and the technology as illustrated in Fig. 4.8. This approach can be extended to other sub-circuits in order to determine a set of functional constraints. DELIGHT.SPICE [18] and the FRIDGE [19] tools were the first to take into account these concepts.



Simple CMOS current mirror	Constraint Type	Constraint Expression
	Saturation	$V_{DS1} > V_{GS1} - V_{TH1}$
	Saturation	$V_{DS2} > V_{GS2} - V_{TH2}$
	Length Matching	$L_1 = L_2$
	Width Macthing	$W_2 = K * W_1$
	Minimun Width	$W_1 > W_{min}$
	Minimun Length	$L_1 > L_{min}$
	Minimun Gate Voltage	$\lvert V_{GS1} - V_{TH1} \rvert > V_{Gmin}$
	Minimun Gate Voltage	$\lvert V_{GS2} - V_{TH2} \rvert > V_{Gmin}$

Fig. 4.8 Functional constraints on a CMOS current mirror

In summary, the sizing rules methodology imposes some constraints not only in the *design space*, formed by the device sizes, but also in the *functional space*. In this context, the search space is decomposed in design space and functional space as illustrated in Fig. 4.9.

In a traditional optimization approach there is a mapping between a point in design space *d* (Fig. 4.9a) and a set of performances in performance space, *p* (Fig. 4.9c). In order to find a solution that satisfies the performance and functional constraints, usually, it requires the computation of many points from the design space. The achieved solution may result in a pathological case, in this condition, the result is not feasible (I in Fig. 4.9c). In the same way, the subspace of d defined by the interception of all functional constraints, the functional space f, may also produce *pathological* solutions, this time all functional constraints are satisfied but misses some performances specs (II in Fig. 4.9c). The *feasible region* defines a set

of points in design space that satisfies both the performance constraints, as well as, the functional constraints. The multidimensional subspace of design parameters which fulfills all functional constraints is called in this work, the *feasibility space* (Fig. 4.9b) and the mapping of this space in performance space in called the *feasibility region* (Fig. 4.9c). If the multidimensional *feasible space* is known the computation time can be highly reduced.

Throughout this chapter, a new method which explores the properties of the sizing rules method and learning algorithms is developed in order to build a model for the functional feasibility space, the *feasibility model*. The aim of this approach is to accelerate the search towards the *performance feasible region* ensuring a proper operation of the analog circuits.

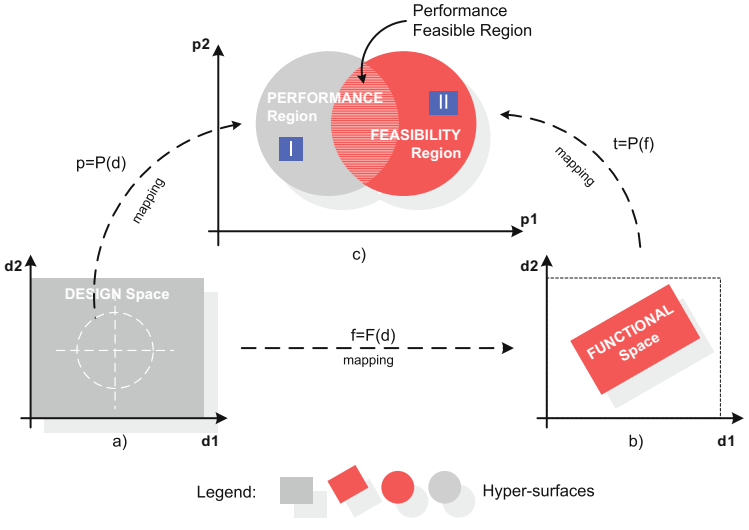


Fig. 4.9 Abstraction of analog circuit feasibility region

4.2.2 Methodology Overview

The evolutionary search algorithms in general have a common behavior. They cyclically generate new moves from the most fitness samples, evaluate them and then discard the less fitness ones. The less fitness offspring information is never used to decide what the next move should be or what path should be followed in getting to a local optimum. Rather than discarding information about the search, this new strategy uses all information from the evolutionary process to help us to make predictions about new data and improve the efficiency of the search algorithm. The new GA-SVM approach incorporates a learning model in the GA optimization cycle based on SVMs. The original GENOM optimization architecture is expanded with modeling capabilities as illustrated by Fig. 4.10.

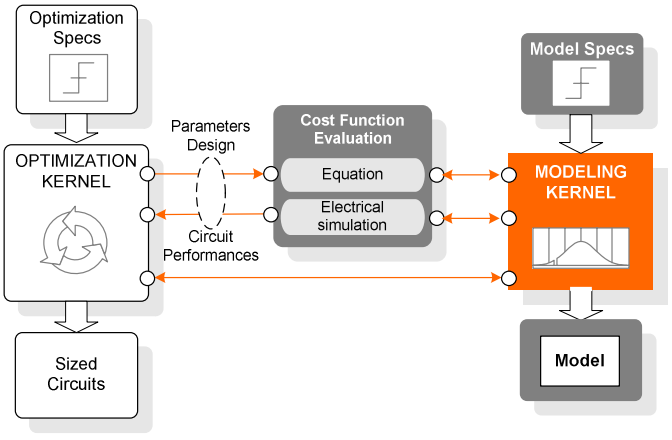


Fig. 4.10 Optimization-Based methods architecture

The learning scheme of analog circuit design is now composed by the interaction of two computational machines, the GA search optimizer and the SVM learning engine. Fig. 4.11 illustrates the block diagram for the optimization kernel with learning algorithm. The SVM models can influence the overall evolutionary process efficiency in two ways.

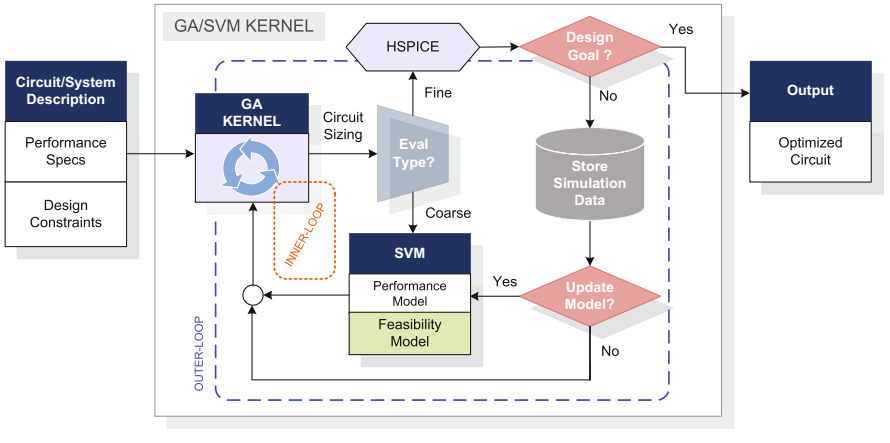


Fig. 4.11 Block diagram of the GA-SVM algorithm

When used as the performance model, the regression model establishes the mapping between the design variables and the performance parameters. This allows their combination to produce an approximation of the fitness function [20], as illustrated in Fig. 4.12, which is used to replace the expensive SPICE-like evaluations in the GA cycle. Potentially, this approach decreases the number of expensive true fitness evaluations and allows a better convergence rate.

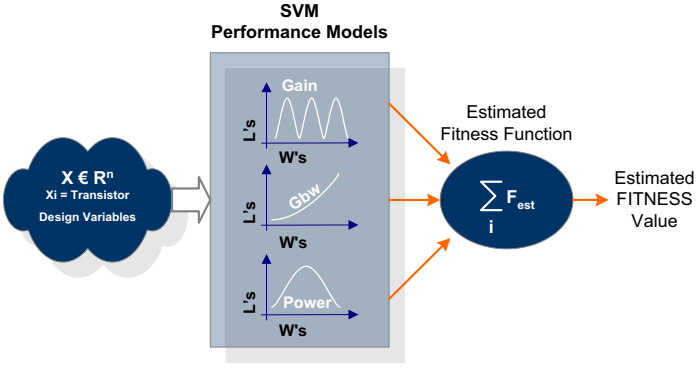


Fig. 4.12 Estimated fitness function with SVM performance model

However, the SVM model presented throughout this work is defined as a two class classifier model. The objective is to estimate the most promising regions, from the design space, to be explored. With this knowledge, the selection method will decide those solutions that will be accepted to proceed on the evaluation process, and those that will be rejected, because they are out or far from one of the most promising regions. The gain in this case is the number of avoided fine evaluations (normally heavy time-consuming electrical simulations) in each generation.

4.2.3 The Feasibility Model Formulation

The GENOM SVM feasibility model is built as a two class classifier model one single time, before evolution cycle, using a set of training samples and the discriminate function given by the basic designer rules formulation of expression (4.1). This representative formulation, usually applied in analog circuit design, is utilized to define the contour of the feasibility region of the feasibility model of SVM. Those solutions satisfying designer rules belong to the class of *feasible region*, and the set of other ones form the *infeasible region*. Solutions are labeled as *feasible* or *infeasible* solutions accordingly the region they belong to. Thus, the feasibility design space is defined by the geometry constrained posed in the range of the design sizes and the functional constraints imposed mainly by the circuit designer rules such as overdrive voltages with some margins, illustrated in Fig. 4.13.

$$VGS > VT + 50mV \quad and \quad VDS > VDSAT + 50mV \tag{4.1}$$

To illustrate this concept in R^2, let us consider a simple Active RC low pass filter with gain *A0*, and frequency *f0* illustrated in Fig. 4.14. The feasibility contour is drawn with respect to capacitor *C1* an *R2*.

The feasibility model $M_f(x)$ defines a function that estimates the front-end between the feasible and infeasible space delimited by the geometry domain. The feasible sub-space, normally a small percentage of the total search space, is built

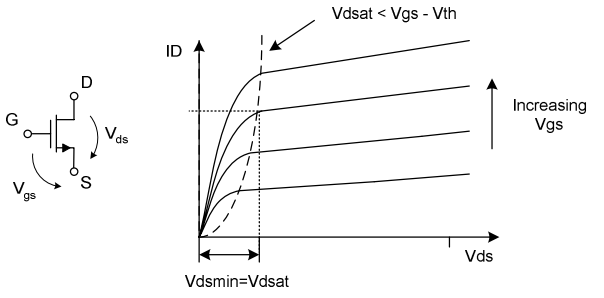


Fig. 4.13 Ids-Vds characteristic of short channel NMOS transistor

by the next sequence of actions. Taking the geometry constraints of the problem, each variable range is divided in equidistant points and is then evaluated by the circuit simulator.

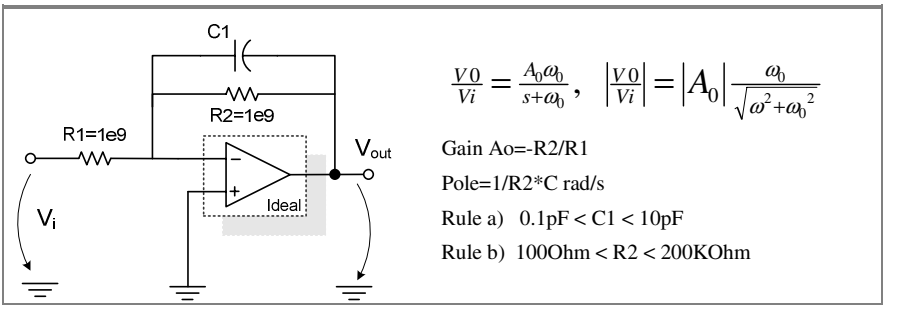


Fig. 4.14 Active RC filter

Another alternative is to sample a number of points proportional to the size of the design space. Next, they are classified in two data sets, the feasible data that satisfies all the designer rules and infeasible data, data which does not satisfy the design rules or was derived from convergence problems. Then, the samples are used as the train sequence to obtain the SVM classification model. The same HSPICE simulations used to build the feasibility model were reused to get the performance measures to train and build the SVM performance model for each performance parameter.

4.2.4 SVM Model Generation and Improvement

In order to improve the success and performance of the SVM feasibility model two enhancements were included in the model, a data sampling with parameter normalization preceded by an unbalanced data management mechanism.

The training data samples were previously evaluated by electric simulations, using a grid search structure which performs data normalization on the design variables using the scale [-1…+1] to prevent the formation of biasing models as explained in 4.2.5. Then, this process is followed by a pre-processing handling phase aiming to balance the data samples from the two main classes, the feasible and infeasible region. Due to the high number of constraints in analog design circuits and large design space available, only a very small region belongs to the interesting class making more difficult the classifier task. The techniques proposed to handle the problem include a novel 3-step stratified method to oversample and undersample the training data set. The objective is to collect the right subset of data samples from the pool of evaluated grid samples in order to build an efficient and accurate feasibility model. The implementation details are given next.

4.2.5 Handling Unbalanced Data in Circuit Designs

Unbalanced data problems impose some difficulties to the classifier task [12],[21-22]. The main pointed reasons are that most current classifier systems like SVM tend to optimize the overall accuracy without considering the weight of relative distribution of each class and they are designed to generalize from sample data to avoid the noise. The GENOM SVM kernel addresses the unbalanced design problems by automatically employing a novel 3-step sampling mechanism adjusted to analog circuit design. First, it implements an over-sampling in the infeasible region in order to increase the samples of the minority class, next refine the frontier between the feasible and infeasible region and finally in the third step, reduce the majority class, by removing those samples far away from the feasibility regions. The estimated effect is illustrated in Fig. 4.15.

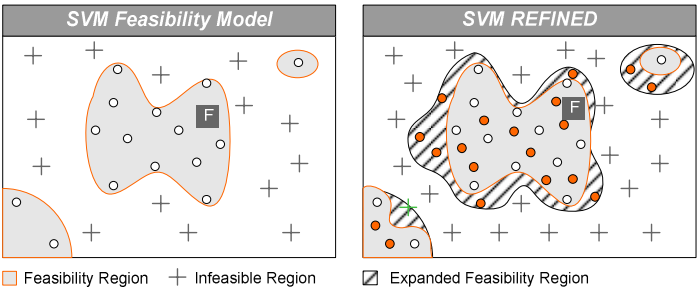


Fig. 4.15 Expected balance effect in design space

To accomplish these tasks, a first sampling strategy based on the classical grid search method as described in 3.3.2 is applied in first place. By default, during the evaluation phase a lot of statistics information is collected for each sample data among which the number of positive and negative samples for each class, the

number of constraints satisfied given by the designer rules satisfied for each sample, the measure of constraints violation and so on. In addition, three new data sets were created: *Fs* embraces the set of evaluated samples that satisfies all constraints (feasible region), *Bs* appends the subset of sample data in boarder region (satisfies all constraints except one, two or three) and finally, *Is* attaches the remaining sample data in the infeasible region and sorts in ascending order of constraint violation value. Fig. 4.16, illustrates the idea of the search space subdivision into feasible and infeasibility regions.

The design experience acquired during this research, in several case studies, has shown that the ratio between positives and negatives samples is in order 0.04 to 0.07 for a total of 2000 uniform random points. An attempt to build a SVM feasibility model under such unbalanced data should result in low and biased performance models. To improve the estimation rate of the feasibility regions in new data, and to increase the efficiency of the model for more complex problems, two new sampling strategies were applied. First, oversampling the data of set *Fs* and Bs by random mutation in vicinity of the original data ("ball" mutation) and second, undersampling the elements of set *Is* by a factor equal to the unbalanced ratio, discarding always the last samples of the set.

Then, in the third step, a balanced SVM two class classifier model is finally built with the train data set, *Ts* being the union of the three final sets *Fs*, *Bs* and *Is* ($Ts = Fs \cup Bs \cup Is$) where the set of positive samples is given by *Fs*+*Bs* and *Is* is associated with the negative set. After that, the model is used to further generate new interior points of the feasibility region and neighborhood. Only the samples classified as positives will be evaluated by the true fitness function. In the end, the model is updated for the last time and the job is returned to the main process where it will be used together with an evolutionary algorithm to find the solution to the analog circuit design problem.

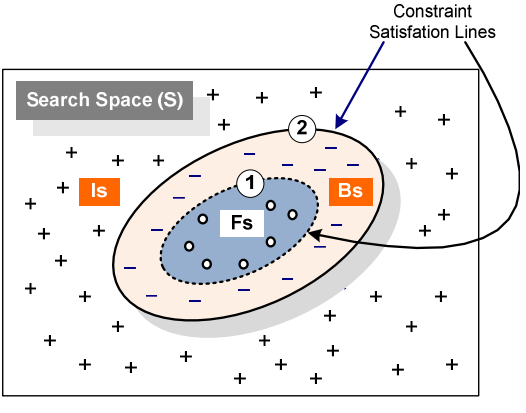


Fig. 4.16 Stratified vision of the search space by feasibility regions

4.2.6 GA-SVM Optimization Overview

The new GA-SVM approach uses all information acquired from the evolutionary process in order to make predictions about new data and improve the efficiency of the search algorithm.

The initialization phase of the GA algorithm is replaced by the sampling mechanism and model generation described in the two earlier sub-sections. Then, the evolutionary algorithm follows the sequential GA optimization algorithm with the exception of the evaluation phase. Here, the evaluation phase is preceded by an *active learning phase,* which uses the feasibility information from the model to decide which of the new offspring will be accepted, to proceed on the evaluation process and those that will be rejected from evaluation because they are out or far away from the most promising regions. The present approach uses a heavy time-consuming electrical simulator to evaluate the true fitness function for each submitted chromosome. Thus, the number of avoided fine evaluations identified by the active learning module in each generation represents a gain of efficiency of this approach and justifies one of the requirements of this implementation. The active evaluation process also implements an aggressive local search around the best individuals in the population, when the number of individuals selected by the active learning module is low. The block diagram of GA-SVM algorithm is illustrated in Fig. 4.17.

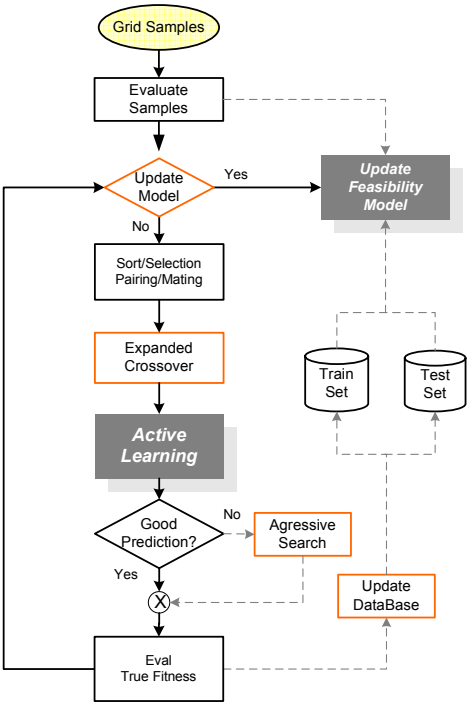


Fig. 4.17 Data flow of GA-SVM algorithm

4.2.7 Comments on the Methodology

This section clarifies the options taken by the presented methodology. To begin with, the constraint stratified vision (Fig. 4.16) used to deal with the unbalanced data problem of analog circuits design applications was implemented to pursue one the SVM fundamental principles, which says that only the support vectors contribute to the decision rule. To build a SVM model efficiently, using the constraint satisfied approach, it is only needed to manage the minority class set and the "best" infeasible samples because that is where the support vectors are present. This way, the management of the huge set of infeasible samples (majority class set) became simplified.

The choice of the training samples and the necessary initial grid resolution used to generate the SVM model, has a great impact on the quality of the model, and will affect the final model performance for unknown data. In the lack of a universal answer to this question, the approach taken in this research follows a simple rule, based on the percentage of the total search space and on the following belief: the feasibility model embodies the circuit's operational zone not in a single but in several points, satisfying or not the problem's specifications. The specs do not affect circuit's feasibility. In the first global sampling, the grid resolution should be chosen in such a way that, at least, one or several feasible points for each operational region should be detected or at least a reasonable number of positive sample points should be collected. If this condition is not met, the following measures can be taken: (a) increase the number of samples, by default it has the same effect of increasing the sampling resolution; (b) relax the contour of the feasible region, that is, accept in the feasible region those infeasible samples close to the feasible region; and/or (c) relax the constraints of the problem, this case needs user involvement.

4.3 Conclusions

The requirements of modern analog design automation tools are placing an increasing emphasis on analytic capabilities. Data mining technology has become an essential instrument in the analysis of large volumes of data in several activity domains. This chapter reviewed a SVM learning machine implementation applied to analog circuit optimization. SVM is considered one of the most efficient techniques belonging to the class of machine learning algorithm able to infer knowledge from data samples. This knowledge is useful to make predictions about new data or to get a better understanding of the system that generated the data. However, to manipulate an SVM tool with an acceptable level of usability and performance, four main tasks should be addressed: data normalization, data balancing, optimal parameters selection and data validation. The influence of these design decisions were illustrated by well known examples.

References

[1] Mitchell, T.: Machine learning. McGraw-Hill, New York (1997)

[2] Edelstein, H.A.: Introduction to data mining and knowledge discovery. In: Two Crows Corporation, 3rd edn. (2003), http://www.twocrows.com/intro-dm.pdf (Accessed March 2009)

[3] Anderson, D., Neil, G.M.: Artificial neural networks technology - A DACS state-of-the-art report. Tech Rep. Data and Analysis Centre for Software (1992)

[4] Jain, A.K., Mao, J., Mohiuddin, K.M.: Artificial neural networks: A tutorial. IEEE Computer, 31–44 (1996)

[5] Kroose, B.J., Smagt, P.V.: An introduction to neural networks. University of Amsterdam (1993), http://citeseer.ist.psu.edu/kroose93introduction.html (Accessed March 2009)

[6] Moore, A.W.: Regression and classification with neural networks. School of Computer Science Carnegie Mellon University (2006), http://www.cs.cmu.edu/~awm (Accessed March 2009)

[7] Farina, M.: A neural network based generalized response surface multiobjective evolutionary algorithm. In: Proc. Congress on Evolutionary Computation, vol. 1, pp. 956–961 (2002)

[8] Gaspar-Cunha, A., Vieira, A.: A Multi-Objective Evolutionary Algorithm Using Neural Networks To Approximate Fitness Evaluations. International Journal of Computers, Systems and Signals 6(1), 18–36 (2005)

[9] Burges, C.J.: A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery 2(2), 121–167 (1998) http://www.umiacs.umd.edu/~joseph/ (Accessed March 2009)

[10] Moore, A.: K-means and hierarchical clustering - tutorial slides. School of Computer Science Carnegie Mellon University (2006), http://www.autonlab.org/tutorials/kmeans.html (Accessed March 2009)

[11] Nilsson, N.J.: Introduction to machine learning - an early draft of a proposed textbook. Department of Computer Science. Stanford University, Stanford (1996) http://robotics.stanford.edu/~nilsson/ (Accessed March 2009)

[12] Milenova, B.L., Yarmus, J.S., Campos, M.M.: SVM in oracle database 10g: Removing the barriers to widespread adoption of support vector machines. In: Proc. 31st International Conference on Very Large Data Bases, pp. 1152–1163 (2005)

[13] Smedt, B., Gielen, G.: WATSON: Design space boundary exploration and model generation for analog and RFIC design. IEEE Trans. Computer-Aided Design of Integrated Circuits and Systems 22(2), 213–224 (2003)

[14] Zizala, S., Eckmuller, J., Graeb, H.: Fast calculation of analog circuits feasibility regions by low level functional measures. In: Proc. Int. Conf. on Electronics, Circuits and Systems, pp. 85–88 (1998)

[15] Gräb, H., Zizala, S., Eckmüller, J., Antreich, K.: The sizing rules method for analog inte-grated circuit design. In: IEEE/ACM International Conference on Computer-Aided Design, pp. 343–349 (2001) doi:10.1109/ICCAD.2001.968645

[16] Wolfe, G.A.: Performance macro-modeling techniques for fast analog circuit synthesis. Ph.D. dissertation, Dept. of Electrical and Computer Engineering and Computer Science, College of Engineering, University of Cincinnati, USA (1999)

[17] Schwencker, R., Eckmueller, J., Graeb, H., Antreich, K.: Automating the sizing of analog CMOS circuits by consideration of structural constraints. In: Proc. Design, Automation and Test in Europe Conference and Exhibition, pp. 323–327 (1999)
[18] Nye, W., Riley, D.C., Sangiovanni-Vincentelli, A., Tits, A.L.: DELIGHT.SPICE: An optimization-based system for the design of integrated circuits. IEEE Trans. Computer-Aided Design 7(4), 501–519 (1998)
[19] Medeiro, F., et al.: A Statistical optimization-based approach for automated sizing of analog cells. In: Proc. ACM/IEEE Int. Conf. Computer-Aided Design, pp. 594–597 (1994)
[20] Kiely, T., Gielen, G.: Performance modeling of analog integrated circuits using least-squares support vector machines. In: Proc. Design, Automation and Test in Europe Conference and Exhibition, vol. 1, pp. 448–453 (2004)
[21] Liu, A.: The effect of oversampling and undersampling on classifying imbalanced text datasets, Master thesis, University of Texas, USA (2004)
[22] Romano, R.A., Aragon, C.R., Ding, C.: Supernova recognition using support vector machines. In: Proc. 5th International Conference on Machine Learning and Applications, pp. 77–82 (2006)