J Glob Optim (2012) 53:297–315
DOI 10.1007/s10898-011-9692-3

Asset portfolio optimization using support vector machines and real-coded genetic algorithm

Pankaj Gupta · Mukesh Kumar Mehlawat · Garima Mittal

Received: 2 February 2011 / Accepted: 7 February 2011 / Published online: 22 February 2011

Abstract This paper presents an integrated approach for portfolio selection in a multicriteria decision making framework. Firstly, we use Support Vector Machines for classifying financial assets in three pre-defined classes, based on their performance on some key financial criteria. Next, we employ Real-Coded Genetic Algorithm to solve a mathematical model of the multicriteria portfolio selection problem in the respective classes incorporating investor-preferences.

Keywords Portfolio optimization · Support vector machines · Real-coded genetic algorithm · Multicriteria decision making

1 Introduction

The asset portfolio selection involves obtaining optimal proportions of the assets for constructing a portfolio that respects investor-preferences. Portfolio selection as a field of study began with the Markowitz model [30] in which return is quantified as the mean and risk as the variance. Konno and Yamazaki [25] used absolute deviation and Speranza [39] used semi-absolute deviation to measure risk in portfolio selection. In these studies of portfolio selection, return and risk are considered as the two fundamental factors that govern investors' choice. However, it is often found that not all the relevant information for portfolio selection can be captured in terms of return and risk only. The other considerations/criteria might be of equal, if not greater, importance to investors. By considering these in the portfolio selection model, it may be possible to obtain portfolios in which a deficit on account of the return and risk criteria is more than compensated by portfolio performance on other criteria, resulting in greater overall satisfaction for investors. Thus, the multicriteria portfolio selection models

P. Gupta · M. K. Mehlawat · G. Mittal
Department of Operational Research, University of Delhi, Delhi, India

P. Gupta (✉)
Flat No.-01, Kamayani Kunj, Plot No. 69, Indraprastha Extension, Delhi 110092, India
e-mail: pgupta@or.du.ac.in

have received great interest from researchers in the recent past, e.g. the models studied by Arenas et al. [2], Ehrgott et al. [12], Fang et al. [14], Gupta et al. [15,16]. One can also refer to [1,3,31,33,34] for use of optimization models in financial decision making. In the financial investment markets, several different assets, such as stocks, bonds, foreign exchanges, options, real estates and future contracts are available for trading. It may be noted that different assets may be showing distinct characteristics vis-à-vis the financial criteria. Given that not all the assets in the market would be appropriate for any one given investor, it would be desirable that these assets be stratified into different classes on the basis of some pre-defined characteristics. Further, based on investor-preferences, one needs to select some good quality assets from a given class to build an optimum portfolio. In the real-world scenario, financial advisors and investment companies use various techniques to profile investors and then recommend a suitable set of assets from which an optimal portfolio is built.

In view of the foregoing discussions, it becomes necessary to identify an appropriate technique for asset classification. In this paper, we use Support Vector Machines (SVM) for asset classification. SVM is a machine-learning technique based on statistical learning theory [41]. An important property that made SVM a promising tool is its implementation of Structural Risk Minimization [41] which aims to minimize a bound on the generalization error rather than on the empirical error. It attempts to construct an optimal separating hyperplane by transforming a nonlinear object into a high dimension feature space and thus gives a good generalization performance on a wide range of problems, such as text categorization [21], pattern recognition [35] and bioinformatics [44]. Interested readers may refer to [4,8] for details on SVM. The SVM approach has also been used in several financial applications such as stock selection, credit rating, time series prediction, insurance claim fraud detection, corporate credit rate prediction and bankruptcy prediction, see [13,19,23,27,38,40,42].

The focus of the present research is to develop a hybrid approach to facilitate the investors in investment decision making. We consider three asset classes as in Gupta et al. [16], based on three financial evaluation indices, namely, return, risk and liquidity. We use SVM with radial basis function kernel to classify the chosen sample of financial assets into these classes. Since parameter selection plays a crucial role in high prediction accuracy and stability of the classifier, we employ a grid search technique using 10-fold cross validation to find the optimal parameter values in the SVM. We, then implement Real Coded Genetic Algorithm (RCGA) in the respective classes to build optimal portfolio based on four financial criteria, namely, short term return, long term return, risk and liquidity. A brief discussion of these criteria follows.

For the portfolio return, we consider short term return (average performance of the assets during the 12-month period) and long term return (average performance of the assets during the 36-month period). This is done in order to capture the subjective preferences of the investors for portfolio return. For a given expected return, the investor penalizes negative semi-absolute deviation which is defined as portfolio risk. Liquidity is considered in terms of the probability of conversion of an investment into cash (turnover) without any significant loss in value.

This paper is organized as follows. In Sect. 2, we present a brief introduction of SVM and Genetic Algorithms. We also discuss the mathematical model of the multicriteria portfolio selection problem in this section. In Sect. 3, we present an empirical study using 36-month data series in respect of 150 assets listed on the National Stock Exchange (NSE), Mumbai, India. This section also includes a discussion of the results obtained. Finally in Sect. 4, we furnish our concluding remarks.

2 Preliminaries and notations

In this section, we present necessary details of SVM and Genetic Algorithms (GA) to brief the readers about these techniques. We also discuss the multicriteria portfolio selection model developed by Gupta et al. [15] which is used for portfolio optimization in this paper.

2.1 Brief overview of support vector machines

We briefly describe the basic SVM concepts for typical two-class classification problems. These concepts can also be found in [4,8]. In SVM approach, the main aim of an SVM classifier is to determine the decision boundary or hyperplane that optimally separates two classes of input data points. Given a training set of instance-label pairs $(x_i, y_i), i = 1, 2, \ldots, m$ where $x_i \in R^n$ and $y_i \in \{-1, +1\}$, for the linearly separable case, the data points will be correctly classified by

$$\langle w.x_i \rangle + b \geq +1 \quad \text{for} \quad y_i = +1. \tag{1}$$

$$\langle w.x_i \rangle + b \leq -1 \quad \text{for} \quad y_i = -1. \tag{2}$$

We can combine (1) and (2) into the following set of inequalities:

$$y_i(\langle w.x_i \rangle + b) - 1 \geq 0 \quad \forall\, i, \tag{3}$$

where w is the normal vector of the hyperplane and b is the bias value. The SVM finds an optimal separating hyperplane with the maximum margin by solving the following optimization problem:

$$\min_{w,b} \frac{1}{2} w^T.w \quad \text{subject to } y_i(\langle w.x_i \rangle + b) - 1 \geq 0 \quad \forall\, i. \tag{4}$$

The above quadratic optimization problem can be solved by finding the saddle point of the Lagrange function L_P

$$L_P(w, b, \alpha) = \frac{1}{2} w^T.w - \sum_{i=1}^{m} \alpha_i (y_i(\langle w.x_i \rangle + b) - 1) \tag{5}$$

where $\alpha_i \geq 0; i = 1, 2, \ldots, m$ are the Lagrange multipliers. The search for an optimal saddle point is necessary because the function L_P must be minimized with respect to the primal variables w and b and maximized with respect to the non-negative dual variables α_i. The Karush-Kuhn-Tucker (KKT) optimality conditions can be used for transforming L_P to the dual Lagrangian $L_D(\alpha)$

$$\begin{aligned} &\max_{\alpha} L_D(\alpha) = \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i.x_j \rangle \\ &\text{subject to} \quad \alpha_i \geq 0, \quad i = 1, \ldots, m \text{ and } \sum_{i=1}^{m} \alpha_i y_i = 0. \end{aligned} \tag{6}$$

To find the optimal hyperplane, the dual Lagrangian $L_D(\alpha)$ must be maximized with respect to non-negative α_i. This is a standard quadratic optimization problem that can be solved by using some standard optimization method. The solution α_i for the dual optimization problem determines the parameters w^* and b^* of the optimal hyperplane. Thus, we

obtain an optimal decision hyperplane $f(x, w^*, b^*)$ given by Eq. (7) and an indicator decision function sign $[f(x, w^*, b^*)]$.

$$f(x, w^*, b^*) = \sum_{i=1}^{m} y_i \alpha_i^* \langle x_i . x \rangle + b^*. \tag{7}$$

The above concepts can also be extended to the nonseparable case. The goal is to construct a hyperplane that makes the smallest number of errors. For this purpose, we introduce the non-negative slack variables $\xi_i \geq 0,\ i = 1, \ldots, m$ such that

$$\langle w . x_i \rangle + b \geq +1 - \xi_i \quad \text{for} \quad y_i = +1. \tag{8}$$

$$\langle w . x_i \rangle + b \leq -1 + \xi_i \quad \text{for} \quad y_i = -1. \tag{9}$$

If errors happen to the classification of training data, ξ_i will be larger than zero. Therefore, a lower $\sum_{i=1}^{m} \xi_i$ is preferred when determining the separating hyperplane. For this purpose, a penalty parameter $C > 0$ is added to control the allowable error ξ_i. The objective function then changes to

$$\begin{aligned} &\min_{w,b,\xi} \ \frac{1}{2} w^T . w + C \sum_{i=1}^{m} \xi_i \\ &\text{subject to} \quad y_i (\langle w . x_i \rangle + b) + \xi_i - 1 \geq 0, \quad \xi_i \geq 0,\ i = 1, \ldots, m. \end{aligned} \tag{10}$$

This optimization model can be solved using the Lagrangian method, which is almost equivalent to the method for solving the optimization problem in the separable case. Thus, we maximize the Lagrangian $L_D(\alpha)$ as in the separable case,

$$\begin{aligned} \max_{\alpha} L_D(\alpha) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j \langle x_i . x_j \rangle \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m, \\ & \sum_{i=1}^{m} \alpha_i y_i = 0. \end{aligned} \tag{11}$$

The penalty parameter C, which is now the upper bound on α_i, is determined by the user. Finally, the optimal decision hyperplane is same as Eq. (7).

When a linear boundary is inappropriate, the nonlinear SVM can map the input vector into a high dimensional feature space via a mapping function Φ, which is also called kernel function. In the dual Lagrange [see (6)], the inner products are replaced by the kernel function as follows:

$$\langle \Phi(x_i) . \Phi(x_j) \rangle = K(x_i, x_j). \tag{12}$$

Now, the non-linear SVM dual Lagrangian $L_D(\alpha)$ is obtained as

$$\begin{aligned} \max_{\alpha} L_D(\alpha) &= \sum_{i=1}^{m} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{m} \alpha_i \alpha_j y_i y_j K(x_i, x_j) \\ \text{subject to} \quad & 0 \leq \alpha_i \leq C, \quad i = 1, \ldots, m, \\ & \sum_{i=1}^{m} \alpha_i y_i = 0. \end{aligned} \tag{13}$$

This optimization model can be solved using the method for solving the optimization model in the separable case. Therefore, the optimal hyperplane has the following form:

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{m} y_i \alpha_i^* \langle \Phi(x_i).\Phi(x) \rangle + b^* = \sum_{i=1}^{m} y_i \alpha_i^* K(x_i, x) + b^*. \quad (14)$$

Depending upon the choice of kernel, the bias b can form an implicit part of the kernel function. Therefore, if a bias term can be accommodated within the kernel function, the nonlinear Support Vector Classifier can be shown as:

$$f(x, \alpha^*, b^*) = \sum_{i=1}^{m} y_i \alpha_i^* \langle \Phi(x_i).\Phi(x) \rangle = \sum_{i=1}^{m} y_i \alpha_i^* K(x_i, x). \quad (15)$$

Some commonly used kernel functions [4] include polynomial, radial basis function (RBF) and sigmoid kernel, which are given as under.

Polynomial kernel:

$$K(x_i, x_j) = (1 + x_i.x_j)^d \quad (16)$$

RBF kernel:

$$K(x_i, x_j) = \exp(-\gamma \parallel x_i - x_j \parallel^2) \quad (17)$$

Sigmoid kernel:

$$K(x_i, x_j) = \tanh(k x_i.x_j - \delta) \quad (18)$$

In order to improve classification accuracy, the kernel parameters in the above kernel functions should be properly chosen.

2.2 Brief overview of genetic algorithms

GA, a general adaptive optimization search methodology [9,17,18] based on a direct analogy to Darwinian natural selection and genetics in biological systems, is a promising alternative to conventional heuristic methods. GA work with a set of candidate solutions called population. Based on the Darwinian principle of survival of the fittest, the GA obtains the optimal solution after a sequence of iterative computations. Associated with the characteristics of exploitation and exploration search, GA can deal with large search spaces efficiently, and hence has less chance to get local optimal solution. The GA begins with a set of solutions which are represented by chromosomes. The chromosomes from one population are taken and used to form a new population. Each of chromosomes of this new population is called as offspring. The main aim of GA is that the new population should be better than the old one. A fitness function assesses the quality of the solution in an evaluation step. The crossover and mutation operators are the main functions that randomly impact the fitness value. Chromosomes are selected for reproduction by evaluating the fitness value. The fitter chromosomes have higher probability to be selected into the recombination pool using the standard selection techniques.

Figure 1 illustrates the genetic operators of crossover and mutation. Crossover, the critical genetic operator that allows new solution regions in the search space to be explored with some probability of crossover, is a random mechanism for exchanging genes between two

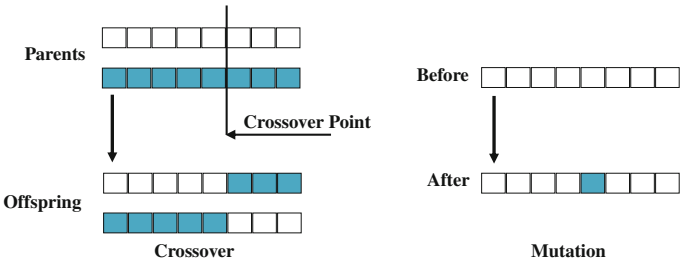
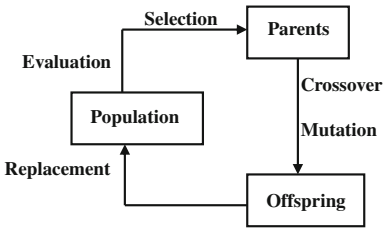


Fig. 1 Crossover and mutation operation

Fig. 2 Evolutionary cycle



chromosomes using any of the available crossover operators such as one point crossover, two point crossover and uniform crossover. In mutation, the genes may occasionally be altered, for example, changing randomly with some mutation probability the gene value from 0 to 1 or vice versa in a binary coded chromosome. This has the potential effect of introducing good gene values that may not have occurred in initial population or which were eliminated during earlier iterations.

Offspring replaces the old population using the elitism or diversity replacement strategy and forms a new population in the next generation. The evolutionary process operates for many generations until termination condition is satisfied. Figure 2 depicts the cycle of GA evolutionary process.

2.3 Multiobjective programming model for asset portfolio selection

We consider the multiobjective portfolio selection model (MOP) developed by Gupta et al. [15].

$$
\begin{aligned}
\text{(MOP)}\quad & \max f_1(x) = \sum_{i=1}^{n} r_i^{12} x_i \quad \text{(Short term return)} \\
& \max f_2(x) = \sum_{i=1}^{n} r_i^{36} x_i \quad \text{(Long term return)} \\
& \min f_3(x) = \sum_{t=1}^{T} \frac{\left|\sum_{i=1}^{n}(r_{it} - r_i)x_i\right| + \sum_{i=1}^{n}(r_i - r_{it})x_i}{2T} \quad \text{(Risk)} \\
& \max f_4(x) = \sum_{i=1}^{n} L_i x_i \quad \text{(Liquidity)}
\end{aligned}
$$

$$\text{subject to} \quad \sum_{i=1}^{n} x_i = 1, \quad \text{(Capital budget constraint)}$$

$$\sum_{i=1}^{n} y_i = h, \quad \text{(Number of assets held in the portfolio)}$$

$$x_i \leq u_i y_i, i = 1, 2, \ldots, n, \quad \text{(Maximal fraction invested in single asset)}$$

$$x_i \geq l_i y_i, i = 1, 2, \ldots, n, \quad \text{(Minimal fraction invested in single asset)}$$

$$x_i \geq 0, i = 1, 2, \ldots, n, \quad \text{(No short selling of assets)}$$

$$y_i \in \{0, 1\}, i = 1, 2, \ldots, n.$$

Here,

r_i the expected return of the ith asset,
x_i the proportion of total fund invested in the ith asset,
y_i the binary variable indicating whether the ith asset is contained in the portfolio or not,

$$y_i = \begin{cases} 1, \text{ if } i\text{th asset is contained in the portfolio} \\ 0, \text{ otherwise} \end{cases}$$

r_i^{12} the average 12-month performance of the ith asset,
r_i^{36} the average 36-month performance of the ith asset,
r_{it} the historical return of the ith asset over the past period t,
u_i the maximal fraction of the capital budget being allocated to the i-th asset,
l_i the minimal fraction of the capital budget being allocated to the i-th asset,
T the total time span.

In the above model, risk is defined as the expected semi-absolute deviation of return of the portfolio below the expected return. For any asset, liquidity is measured with the help of the turnover rate defined as the ratio between the average stock traded at the market and the tradable stock (shares held by public) of that asset. The maximal and minimal fractions of the capital budget allocated to the various assets in the portfolio depends upon a number of factors. For example, one may consider price/value relative of the asset $vis - \acute{a} - vis$ the average of the price/value of all the assets in the chosen portfolio, minimal lot size that can be traded at the market, the past behavior of the price/volume of the asset, information available about the issuer of the asset and trends in the industry of which it is a part. In other words, investors refer to a host of fundamental and technical analysis factors affecting the company and the industry. Since investors differ in their interpretation of the available information, they may allocate the same overall capital budget differently. The constraints corresponding to lower bounds l_i and upper bounds u_i on the investment in individual assets $(0 \leq l_i, u_i \leq 1, l_i \leq u_i, \forall\, i)$ are used to avoid a large number of very small investments (lower bounds) and at the same time to ensure a sufficient diversification of the investment (upper bounds). It may be noted that lower and upper bounds have to be chosen carefully so that the problem does not become infeasible. As for the number of assets held in the portfolio, h indicates number of assets that the investor chooses to include in the portfolio. Of all the assets from a given class, the investor would pick up the ones that are likely to yield the desired satisfaction of his preferences. It is not necessary that all the assets from a given class may configure in the portfolio as well. Investors would differ with respect to the number of assets they can effectively manage in a portfolio.

3 Empirical study

Presented hereunder are the results of an empirical study for which we have relied on a data set of daily closing prices in respect of 150 assets listed on NSE, Mumbai, India, the premier market for financial assets.

3.1 Asset classes

We use the following three classes of assets derived by Gupta et al. [16].

(i) *Class 1: Liquid Assets*
Assets in Class 1 are categorized as liquid assets since mean value for liquidity is the highest in this class. This class is typified by low but widely varying returns.

(ii) *Class 2: High-yield Assets*
Assets in Class 2 are categorized as high-yielding ones, since they have rather high returns. On the expected lines of return/risk relationship, these assets also show high standard deviation. Although, investors may profit from the high returns, they also have to endure the high risk. These assets have low liquidity amongst all the three classes indicating that high-yielding investment involves a longer time horizon.

(iii) *Class 3: Less-risky Assets*
Assets in Class 3 are categorized as less-risky assets, since compared to other classes, these assets have the lowest standard deviation for the class. The return is not high but medium. The liquidity is medium too.

3.2 Classification of assets using SVM

In this paper, LIBSVM software [5] is used to perform multiclass SVM experiments. To allow for multiclass classification, LIBSVM uses one-against-one approach, see [5]. We have split the data into two subsets: a training set of 60% (data of 90 assets) and a testing set of 40% (data of 60 assets) of the total data (data of 150 assets), respectively. We consider three evaluation indices to perform classification, namely, asset returns (average 36-month performance of the assets), standard deviation and liquidity. We first choose the appropriate kernel for implementing SVM. We then choose the penalty parameter C and the kernel parameter. Out of all available kernels for SVM the advantage of using the linear kernel SVM is that there are no parameters to tune except for constant C, but it affects the prediction performance for the cases where the training data is not separable by a linear SVM [11]. For the nonlinear SVM, there is an additional parameter, the kernel parameter, to tune. As discussed in Sect. 2.1, there are three kernel functions for nonlinear SVM, namely, the RBF, the polynomial and the sigmoid kernel. The RBF kernel nonlinearly maps the samples into a higher dimension space unlike the linear kernel, so it can handle the case when the relation between class labels and attributes is nonlinear. Furthermore, the linear kernel is a special case of RBF [22]. In addition, the sigmoid kernel behaves like RBF for certain parameters [29], however, it is not valid under some parameters. The polynomial kernel takes a longer time in the training stage of SVM and it is reported to provide worse results than the RBF kernel, see [19,40]. We, therefore, use the RBF kernel SVM as the default model. There are two parameters associated with the RBF kernel, C and γ. It is not known beforehand which values of C and γ are the best for one problem; consequently, some kind of model selection (parameter search) approach must be employed [20]. This study conducts a gridsearch to find the best values of C and γ using 10-fold crossvalidation. Pairs of (C, γ) are tried and the one with the best cross-validation accuracy is selected. Cross-validation procedure can prevent the overfitting problem. It is well

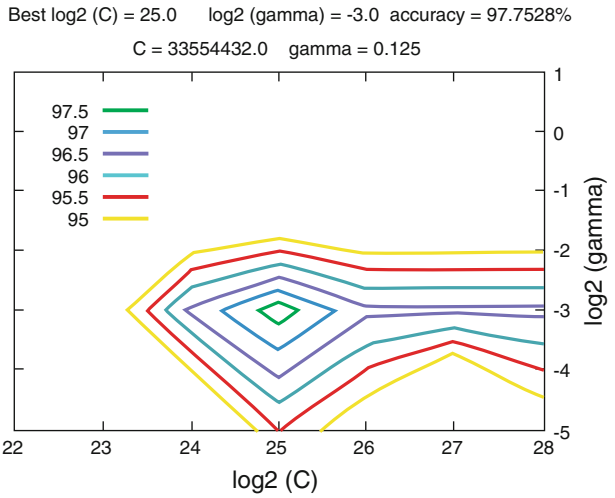


Fig. 3 Grid-search using $C = 2^{22}, 2^{23}, \ldots, 2^{28}$, $\gamma = 2^{-5}, 2^{-3}, \ldots, 2^{1}$

Table 1 The result of grid-search

C	γ			
	2^{-5}	2^{-3}	2^{-1}	2^{1}
2^{22}	93.2584	94.382	93.2584	89.8876
2^{23}	94.382	94.382	93.2584	89.8876
2^{24}	93.2584	96.6292	93.2584	89.8876
2^{25}	95.5056	**97.7528**	93.2584	89.8876
2^{26}	94.382	96.6292	93.2584	89.8876
2^{27}	92.1348	96.6292	93.2584	89.8876
2^{28}	94.382	96.6292	93.2584	89.8876

Table 2 Classification confusion matrix for test data

Actual	Predicted		
	Class 1	Class 2	Class 3
Class 1	20	1	2
Class 2	1	19	1
Class 3	0	0	16
Total	21	20	19

established that trying exponentially growing sequences of C and γ is a practical method to identify good parameters, for example, $C = 2^{22}, 2^{23}, \ldots, 2^{28}$, $\gamma = 2^{-5}, 2^{-3}, \ldots, 2^{1}$. After conducting the grid-search on the training data, we find that the optimal (C, γ) is $(2^{25}, 2^{-3})$ with the cross-validation rate of 97.7528% (see Fig. 3 and Table 1). After obtaining the optimal (C, γ), the SVM classifier is built for the training data. The testing data is then input to the SVM classifier and the prediction accuracy is found to be 91.6666%. The classification confusion matrix is shown in Table 2.

3.3 Using RCGA to solve (MOP)

The financial application of GA is growing with successful applications in trading system [7,10], portfolio selection [6,26,28,36], bankruptcy prediction [24], credit evaluation [43] and budget allocation [32].

The 21 financial assets classified in class 1, 20 financial assets classified in class 2 and 19 financial assets classified in class 3 comprise the population for the three classes. We construct a portfolio comprising 7 assets with the corresponding upper and lower bounds of capital budget allocation. Table 3 provides the data corresponding to the short term return (average 12-month performance of the assets), long term return (average 36-month performance of the assets), risk and liquidity of assets in classes 1, 2 and 3. It may be noted that the average returns used in this study are the average of the averages, that is, the average monthly returns. The monthly returns are based on the daily returns. We use average 36-month performance of the asset as the expected return in the calculations. Liquidity of the assets is measured using the respective turnover rates.

Now, for constructing optimum portfolio for a given investor, one needs to pick the assets from the suitable class as per investor-preferences and find the best combination of assets to be invested in according to the decision model discussed in Sect. 2.3. RCGA has been used to solve the model. We present below the details of the algorithm used.

Chromosome encoding

A gene in a chromosome is characterized by two factors: Locus (i.e. the position of the gene located within the structure of chromosome) and allele (i.e. the value the gene takes). In the proposed encoding method, the position of the gene is used to represent the ID number of the asset and its value is used to represent the weight for constructing the portfolio. An initial population of randomly generated individuals (chromosomes) is created as follows:

- Fix length (n) of each chromosome as the number of assets in the respective class.
- Initially, a random mask which is a binary string of same length as the desired parent chromosome is generated with exactly h number of 1's to handle the constraint corresponding to the number of assets held in the portfolio, i.e.

$$\sum_{i=1}^{n} y_i = h.$$

- The parity of each bit in the mask determines selection or rejection of the corresponding asset. Mask value 1 for a particular ID indicates selection of the corresponding asset, whereas a mask value 0 indicates dropping of the corresponding asset.

Weight generation

In this step of encoding procedure, the value (x_i representing weight) of the gene corresponding to the selected asset (value 1 in the binary mask) in the chromosome is generated randomly within the bounds of the corresponding constraints as follows:

- Maximal fraction of the capital that can be invested in a single asset:

$$x_i \leq u_i y_i, \quad i = 1, 2, \ldots, n.$$

Table 3 Input data for SVM classified assets

Class	Feature	Asset ID Number										
		A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
Class 1	Risk	0.19770648	0.189761167	0.117766661	0.185046005	0.216346183	0.166456688	0.288520008	0.233211042	0.19970412	0.120048255	0.205283045
	Liquidity	0.006109142	0.003591955	0.003095826	0.005696145	0.009368094	0.016794986	0.011493328	0.006338305	0.001846994	0.002438463	0.005702438
	Short term return	−0.06077381	0.17239913	0.027558756	0.056132232	0.066387225	0.068978431	0.181090054	0.05893478	−0.003009473	0.152577637	0.149557092
	Long term return	0.112130599	0.150582192	0.117446593	0.104976919	0.116706662	0.105482741	0.203612013	0.172789874	0.093702882	0.115047247	0.100505604
Class 2	Risk	0.31853189	0.140096999	0.168536314	0.206342287	0.272524097	0.423260927	0.383410529	0.115266017	0.133235527	0.365040392	0.338797581
	Liquidity	0.004894308	0.000316634	0.001258499	0.001236353	0.001338484	0.003518941	0.002405999	0.00104677	0.002346163	0.006310535	0.004146338
	Short term return	0.075069636	0.174993984	0.156113607	0.200955517	0.071389657	0.339791987	0.272900218	0.077168331	0.14934383	0.112777266	0.151081989
	Long term return	0.164566724	0.192779347	0.213655993	0.217107317	0.198191086	0.400863598	0.308311638	0.17448066	0.232027703	0.278575042	0.274765251
Class 3	Risk	0.086823714	0.150592797	0.101497201	0.098064463	0.077157871	0.139065751	0.075726301	0.161180503	0.109795166	0.061179014	0.075869607
	Liquidity	0.000880425	0.005036032	0.002011128	0.002513235	0.000796969	0.004706807	0.000589871	0.005479837	0.002668648	0.001320242	0.004125369
	Short term return	0.114866423	0.099496672	0.006143753	0.142397017	0.093605735	0.091569508	0.113085189	0.158429467	0.154770929	0.109801779	0.102725998
	Long term return	0.149209135	0.135865326	0.166222004	0.14868445	0.143821256	0.168470744	0.14905263	0.104916625	0.194915827	0.159265358	0.120716026

Table 3 continued

Class	Feature	Asset ID Number									
		A12	A13	A14	A15	A16	A17	A18	A19	A20	A21
Class 1	Risk	0.308166989	0.252089986	0.293678252	0.186876252	0.215699795	0.189207994	0.104625363	0.360563244	0.132802153	0.194329622
	Liquidity	0.047986537	0.009385935	0.017436841	0.002242128	0.015965537	0.014498742	0.013335473	0.027485234	0.002664776	0.00080972
	Short term return	0.14446243	0.050449949	0.096975294	0.006503264	0.188386073	0.004889209	0.076294803	0.153970686	0.016375832	–0.035621224
	Long term return	0.155311648	0.156014546	0.081754285	0.103794336	0.149878581	0.092244794	0.101311328	0.225222237	0.125001438	0.144125814
Class 2	Risk	0.207508214	0.351373301	0.132485907	0.422472681	0.237018968	0.139706802	0.314365051	0.1625736	0.297598246	–
	Liquidity	0.001797797	0.001158085	0.000599291	0.001044355	0.00084399	8.64528E-05	0.000468126	0.000435948	0.000421428	–
	Short term return	0.067719598	0.192311188	0.199264465	0.309558052	0.175528738	0.162997824	0.085232015	0.113447453	0.154968702	–
	Long term return	0.171328278	0.14649198	0.179849117	0.347337849	0.296327129	0.181336132	0.229487104	0.179298156	0.367002292	–
Class 3	Risk	0.081014349	0.088933214	0.067402867	0.095700739	0.079773114	0.44632272	0.516919057	0.115070929	–	–
	Liquidity	0.000857023	0.000378645	0.00040511	0.003030622	0.002012008	0.013781236	0.012166107	0.000669601	–	–
	Short term return	0.127892921	0.14324552	0.090956669	0.093193036	0.080360023	0.188722414	0.170201677	0.027938748	–	–
	Long term return	0.12385891	0.135375369	0.191621765	0.135012196	0.13470955	0.282126705	0.253822541	0.139645831	–	–

– Minimal fraction of the capital that can be invested in a single asset:

$$x_i \geq l_i y_i\,, \quad i = 1, 2, \ldots, n.$$

Fitness evaluation

The fitness function must take care of all the desired objectives, thus, making a rational trade-off between minimizing risk, maximizing returns and maximizing liquidity. In our fitness function, the only left over constraint, i.e. the capital budget constraint has been incorporated by levying a penalty P for infeasible chromosomes. Let

$$f_5(x) = \left| \sum_{i=1}^{n} x_i - 1 \right|.$$

The fitness function is designed as:

$$\text{Fitness} = w_1 f_1(x) + w_2 f_2(x) - w_3 f_3(x) + w_4 f_4(x) - P f_5(x)$$

where w_j, $j = 1, 2, 3, 4$ is the weight given to the j-th objective function, highlighting the relative importance of a particular objective in a particular class and

$$P = \begin{cases} 10^4, & \text{if } f_5(x) > 10^{-3}, \\ 0, & \text{otherwise.} \end{cases}$$

The objective is to maximize this fitness function.

Selection

We employ 4-player tournament selection as the selection mechanism. Four individuals are randomly selected and the one with the highest fitness is selected for the mating pool. At each generation, elitism is performed by retaining the fittest individual for the next population.

Crossover operator

Standard crossover operators for binary encoding have a high probability of violating the following constraint of the problem (MOP)

$$\sum_{i=1}^{n} y_i = h.$$

Thus, we use shrinking crossover (SX) operator [6]. SX revises the two-point crossover to exchange the same number of $1's$ (inturn exchanging the actual corresponding gene values of parent chromosomes) in the binary mask of the parent by moving the second crossover point leftward until there are equal number of ones between the two crossover points in both the parents. The algorithm of the shrinking crossover operator is given below:

procedure *SX*$(C1, C2)$
  select two crossover points s & t
  until$(C1$ *&* $C2$ *have equal number of* $1's$ *between s &* $t)$
    t = *t-1*
  enduntil
  for$(i$ = *s to* $t)$

Table 4 Main attributes of the problem instances solved

	Class 1	Class 2	Class 3
Probability of crossover	0.45	0.45	0.45
Probability of mutation	0.07	0.07	0.07
Population Size	100	100	100
Length of a chromosome (Number of assets)	21	20	19
Number of generations	5,000	5,000	5,000
Number of assets to be selected	7	7	7
u_i , $\forall\, i$	0.3	0.3	0.3
l_i , $\forall\, i$	0.08	0.08	0.08
Weights of objectives	$w_1 = 0.2, w_2 = 0.25$ $w_3 = 0.2, w_4 = 0.35$	$w_1 = 0.3, w_2 = 0.35$ $w_3 = 0.2, w_4 = 0.15$	$w_1 = 0.22, w_2 = 0.25$ $w_3 = 0.35, w_4 = 0.18$

temp = C1[i]
C1[i] = C2[i]
C2[i] = temp
endfor
endprocedure

Mutation operator

We use swap mutation by swapping the parity information in the binary mask of chromosome. With some small probability of mutation, we select two positions in a chromosome for swap say r and s. If after the swap, mask bit parity for r is 1 and for s it is 0, regenerate the value for r-th gene between the defined ranges:

$$x_r \leq u_r y_r\,,$$
$$x_r \geq l_r y_r\,,$$

and the value of s-th gene is kept as 0.

In what follows, we present computational results. The main attributes of the problem instances solved are summarized in Table 4.

3.4 Experimental results

Class 1

Corresponding to $w_1 = 0.2$, $w_2 = 0.25$, $w_3 = 0.2$ and $w_4 = 0.35$, we obtain the desired portfolio by solving the model (MOP). It may be noted that class 1 is of liquid assets, therefore, the highest weightage (w_4) is given to liquidity objective.

Class 2

Corresponding to $w_1 = 0.3$, $w_2 = 0.35$, $w_3 = 0.2$ and $w_4 = 0.15$, we obtain the desired portfolio by solving the model (MOP). It may be noted that class 2 is of high-yield assets, therefore, the highest weightages (w_1 and w_2) are given to return objectives.

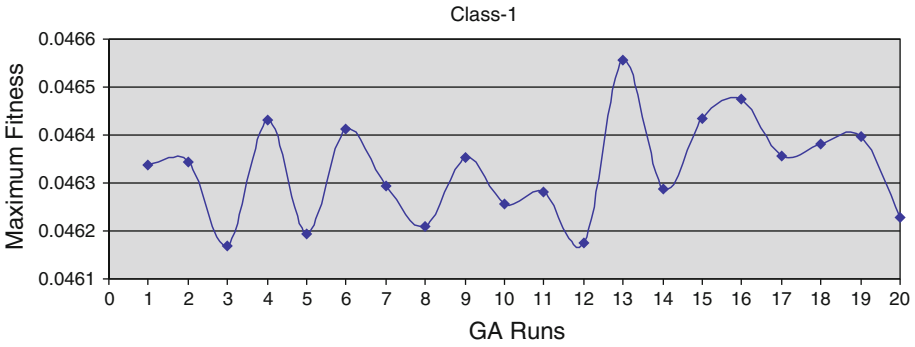


Fig. 4 Maximum fitness vs. GA runs for Class 1

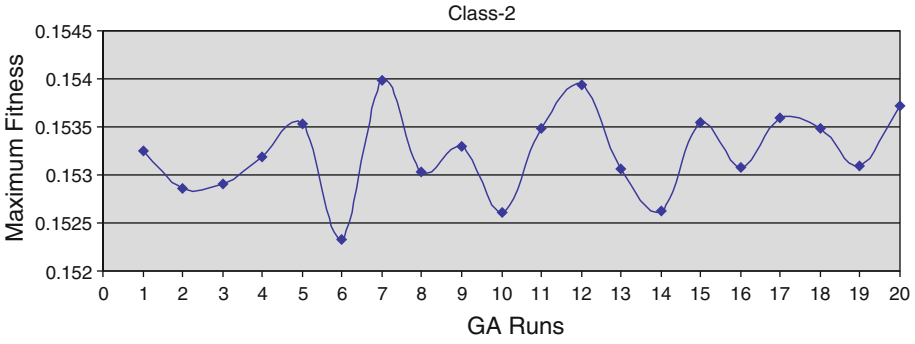


Fig. 5 Maximum fitness vs. GA runs for Class 2

Class 3

Corresponding to $w_1 = 0.17$, $w_2 = 0.23$, $w_3 = 0.45$ and $w_4 = 0.15$, we obtain the desired portfolio by solving the model (MOP). It may be noted that class 3 is of less-risky assets, therefore, the highest weightage (w_3) is given to risk objective.

It may be noted that various multicriteria decision making techniques can be used to determine the weights of the different objective functions. Based upon investor-preferences, Analytical hierarchy process (AHP) [37] is a good technique to determine the weights.

We performed 20 GA runs to check the stability of the solution. Table 5 shows the solution statistics. Figures 4, 5 and 6 shows the sensitivity of the maximum fitness with 20 GA runs for all the three classes. Table 6 provides the attainment values of the various objectives corresponding to best solution out of 20 GA runs for all the three classes. Table 7 present proportions of the assets in the obtained portfolios. A comparison of the solutions for the three classes listed in Table 6 highlights that if investors are looking for high liquidity, they should invest in class 1 assets, i.e. liquid assets. The attainment level of liquidity of the portfolio build from class 1 assets is higher in comparison to class 2 and class 3, but that is accompanied by a medium risk level. If investors are looking for returns, they should invest in class 2 assets, i.e. high-yield assets. The attainment level of returns of the portfolio build from class 2 is higher in comparison to class 1 and class 3, but that carries a higher risk level too. If investors are looking for safe investment, they should invest in class 3 assets, i.e. less-risky assets. The attainment level of risk of the portfolio build from class 3 is lower

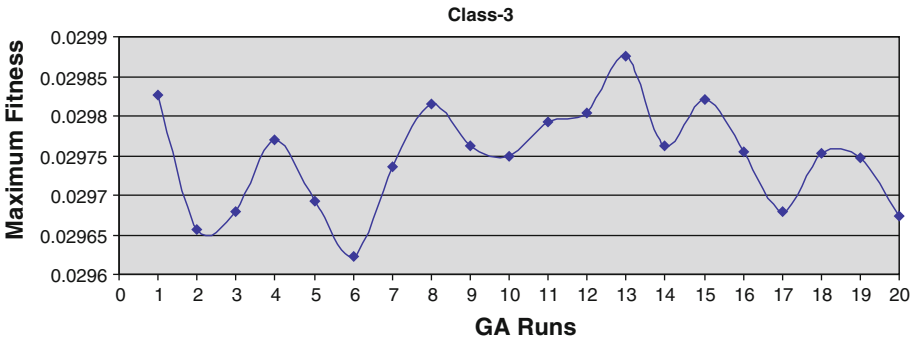


Fig. 6 Maximum fitness vs. GA runs for Class 3

Table 5 Solution statistics for 20 GA runs for the various classes

	Class		
	Class 1	Class 2	Class 3
Best fitness	0.046555	0.153982	0.029875
Average fitness	0.0463282	0.1532298	0.0297486
Standard deviation	0.000105604	0.000439727	6.52028E-05
Coefficient of variation (%)	0.227947185	0.286972182	0.219179529

Table 6 Attainment values of the various objectives

Objective	Class		
	Class 1	Class 2	Class 3
Short term return	0.165134	**0.269611**	0.152307
Long term return	0.181297	**0.340519**	0.202851
Risk	0.190471	0.232001	**0.158369**
Liquidity	**0.017995**	0.002114	0.006025

in comparison to class 1 and class 2, but that supposes accepting medium level of expected returns.

Table 7 Asset allocation

Class	Asset										
	A1	A2	A3	A4	A5	A6	A7	A8	A9	A10	A11
Class 1											
	0	0.081128	0	0	0	0	0.292279	0	0	0.080134	0.081262
Class 2											
	0	0	0	0.083699	0	0.297952	0.087654	0	0	0	0.084653
Class 3											
	0	0	0.080994	0	0.083948	0	0.080611	0.282336	0	0	0

Table 7 continued

Class	Asset									
	A12	A13	A14	A15	A16	A17	A18	A19	A20	A21
Class 1	0.090762	0	0	0	0.085479	0	0	0.289822	0	0
Class 2	0	0	0	0.28384	0.082484	0	0	0	0.080638	–
Class 3	0	0.081813	0.09132	0	0	0.299872	0	0	–	–

4 Conclusions

We have used SVM in this paper to categorize the chosen sample of financial assets into three pre-defined classes, based on three financial evaluation indices, namely, return, risk and liquidity. In each of these classes, RCGA has been used to solve the multiple criteria portfolio selection model considering short term return, long term return, risk and liquidity.

The SVM classifier built in this study promises a high classification accuracy of 91.6666%. The main advantage of the proposed approach is that once such a classifier is obtained, it can then be used to classify any set of randomly chosen assets into the relevant classes. This provide investors a prima facie information about the class of the assets and thus help investors to decide the appropriate investment alternatives. The investors, then, may pick and choose from among these alternatives by obtaining the desired portfolio with the help of RCGA. The advantage of using RCGA is that we need not linearize the risk objective and can solve portfolio selection problem (MOP) in its original form.

Our results indicate that the approach developed here is capable of classifying assets with good accuracy and further is capable of yielding optimal portfolios for each class of assets based on the investor-preferences regarding the financial criteria used.

References

1. AitSahlia, F., Pardalos, P.M., Sheu, Y-C.F.: Optimal execution of time-constrained portfolio transactions. In: Konthoghiorges, E.J., Rustem, B., Winker, P. (eds.) Computational Methods in Financial Engineering, pp. 95–102. Springer, UK (2008)
2. Arenas Parra, M., Bilbao Terol, A., Rodríguez Uría, M.V.: A fuzzy goal programming approach to portfolio selection. Eur. J. Oper. Res. **133**(2), 287–297 (2001)
3. Boginski, V., Butenko, S., Pardalos, P.M.: Mining market data: a network approach. Comput. Oper. Res. **33**(11), 3171–3184 (2006)
4. Burges, C.J.C.: A tutorial on support vector machines for pattern recognition. Data. Min. Knowl. Discov. **2**(2), 955–974 (1998)
5. Chang, C.C., Lin, C.J.: LIBSVM:a library for support vector machines, Software available at http://www.csie.ntu.edu.tw/~cjlin/libsvm (2001)
6. Chen, J.S., Hou, J.L., Wua, S.M., Chang-Chien, Y.W.: Constructing investment strategy portfolios by combination genetic algorithms. Exp. Syst. Appl. **36**(2), 3824–3828 (2009)
7. Colin, A.M.: Genetic algorithms for financial modeling. In: Deboeck, G.J. (ed.) Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets, pp. 148–173. John Wiley, New York (1994)
8. Cristianini, N., Shawe-Taylor, J.: An Introduction to Support Vector Machines. Cambridge University Press, Cambridge, UK (2000)
9. Davis, L.: Handbook of Genetic Algorithms. Van Nostrand Reinhold, New York (1991)
10. Deboeck, G.J.: Using GAs to optimize a trading system. In: Deboeck, G.J. (ed.) Trading on the Edge: Neural, Genetic and Fuzzy Systems for Chaotic Financial Markets, pp. 174–188. Wiley, New York (1994)

11. Drucker, H., Wu, D., Vapnik, V.N.: Support vector machines for spam categorization. IEEE Trans. Neural Network **10**(5), 1048–1054 (1999)
12. Ehrgott, M., Klamroth, K., Schwehm, C.: An MCDM approach to portfolio optimization. Eur. J. Oper. Res. **155**, 752–770 (2004)
13. Fan, A., Palaniswami, M.: Stock Selection Using Support Vector Machines. In: Proceedings of International Joint Conference on Neural Networks **3**, 1793–1798 (2001)
14. Fang, Y., Lai, K.K., Wang, S.Y.: Portfolio rebalancing model with transaction costs based on fuzzy decision theory. Eur. J. Oper. Res. **175**, 879–893 (2006)
15. Gupta, P., Mehlawat, M.K., Saxena, A.: Asset portfolio optimization using fuzzy mathematical programming. Inform. Sci. **178**, 1734–1755 (2008)
16. Gupta, P., Mehlawat, M.K., Saxena, A.: A hybrid approach to asset allocation with simultaneous consideration of suitability and optimality. Inform. Sci. **180**, 2264–2285 (2010)
17. Goldberg, D.E.: Genetic Algorithms in Search Optimization and Machine Learning. Addison-Wesley, Reading, MA (1989)
18. Holland, J.H.: Genetic algorithms. Sci. Am. **267**, 66–72 (1992)
19. Huang, Z., Chen, H., Hsu, C.J., Chen, W.H., Wu, S.: Credit rating analysis with support vector machines and neural networks: A market comparative study. Decis. Support Syst. **37**(4), 543–558 (2004)
20. Hsu, C.W., Chang, C.C., Lin, C.J.: A practical guide to support vector classification. Available at http://www.csie.ntu.edu.tw/~cjlin/papers/guide/guide.pdf (2003)
21. Joachims, T.: Text categorization with support vector machines : Learning with many relevant features. In: Proceedings of the ECML-98, 10th European conference on machine learning, pp. 137– 142 (1998)
22. Keerthi, S.S., Lin, C.J.: Asymptotic behaviors of support vector machines with Gaussian kernel. Neural Comput. **15**(7), 1667–1689 (2003)
23. Khemchandani, R., Jayadeva Chandra, S.: Regularized least squares fuzzy support vector regression for financial time series forecasting. Exp. Syst. Appl. **36**, 132–138 (2009)
24. Kingdom, J., Feldman, K.: Genetic algorithms for bankruptcy prediction. Search Space Research Report No. 01-95, Search Space Ltd. London (1995)
25. Konno, H., Yamazaki, H.: Mean-absolute deviation portfolio optimization model and its applications to the Tokyo Stock Market. Manag. Sci. **37**, 519–531 (1991)
26. Lai, K.K., Yu, L., Wang, S., Zhou, C.: A double-stage genetic optimization algorithm for portfolio selection. In: Proceedings of the 13th international conference on neural information processing: LNCS 4234, pp. 928–937 (2006)
27. Lee, Y.C.: Application of support vector machines to corporate credit rating prediction. Exp. Syst. Appl. **33**(1), 67–74 (2007)
28. Lin, C.M., Gen, M.: An effective decision-based genetic algorithm approach to multiobjective portfolio optimization problem. Appl. Math. Sci. **1**(5), 201–210 (2007)
29. Lin, H.T., Lin, C.J.: A study on sigmoid kernels for SVM and the training of non-PSD kernels by SMO-type methods. Technical report, Department of Computer Science, National Taiwan University (2003)
30. Markowitz, H.: Portfolio selection. J. Finance **7**, 77–91 (1952)
31. Mukuddem-Petersen, J., Mulaudzi, M.P., Petersen, M.A., Schoeman, I.M.: Optimal mortgage loan securitization and the subprime crisis. Optim. Lett. **4**(1), 97–115 (2010)
32. Packard, N.: A genetic learning algorithm for the analysis of complex data. Complex Syst. **4**, 543–572 (1990)
33. Pardalos, P.M., Sandström, M., Zopounidis, C.: On the use of optimization models for portfolio selection: a review and some computational results. Comput. Econ. **7**, 227–244 (1994)
34. Pardalos, P.M., Tsitsiringos, V.: Financial Engineering, Supply Chain and E-commerce. Kluwer, UK (2002)
35. Pontil, M., Verri, A.: Support vector machines for 3D object recognition. IEEE Trans. Pattern Anal. Mach. Intell. **20**(6), 637–646 (1998)
36. Rutan, E.: Experiments with optimal stock screens. In: Proceedings of the 3rd International Conference on Artificial Intelligence Applications on Wall Street, pp. 269–273 (1993)
37. Saaty, T.L.: Fundamentals of Decision Making and Priority Theory with the AHP, 2nd edn. RWS Publications, Pittsburg, PA (2000)
38. Shin, K.S., Lee, T.S., Kim, H.J.: An application of support vector machines in a bankruptcy prediction model. Exp. Syst. Appl. **28**, 127–135 (2005)
39. Speranza, M.G.: Linear programming models for portfolio optimization. Finance **14**, 107–123 (1993)
40. Tay, F.E.H., Cao, L.: Application of support vector machines in financial time series forecasting. Omega **29**(4), 309–317 (2001)
41. Vapnik, V.N.: The Nature of Statistical Learning Theory. Springer, New York (1995)

42. Viaene, S., Derrig, R.A., Baesens, B., Dedene, G.: A comparison of state-of-the-art classification techniques for expert automobile insurance claim fraud detection. J. Risk Insur. **69**(3), 373–421 (2002)
43. Walker, R., Haasdijk, E., Gerrets, M. : Credit evaluation using a genetic algorithm. In: Coonatilake, S., Treleaven, P. (eds.) Intelligent Systems for Finance and Business, pp. 39–59. Wiley, Chichester (1995)
44. Yu, G.X., Ostrouchov, G., Geist, A., Samatova, N.F.: An SVM based algorithm for identification of photosynthesis-specific genome features. In: Proceedings of the Second IEEE Computer Society Bioinformatics Conference, pp. 235–243 (2003)