An Efficient SVM-GA Feature Selection Model for Large Healthcare Databases

Rick Chow[1], Wei Zhong[2], Michael Blackmon[3], Richard Stolz[4], Marsha Dowell[5]
Division of Math and Computer Science[1,2,3]
Johnson College of Business and Economics[4]
School of Nursing[5]
University of South Carolina Upstate
800 University Way
Spartanburg, SC 29303
{rchow[1], wzhong[2], mblackmon[3], rstolz[4], mdowell[5]}@uscupstate.edu

ABSTRACT

This paper presents an efficient hybrid feature selection model based on Support Vector Machine (SVM) and Genetic Algorithm (GA) for large healthcare databases. Even though SVM and GA are robust computational paradigms, the combined iterative nature of a SVM-GA hybrid system makes runtime costs infeasible when using large databases. This paper utilizes hierarchical clustering to reduce dataset size and SVM training time, multi-resolution parameter search for efficient SVM model selection, and chromosome caching to avoid redundant fitness evaluations. This approach significantly reduces runtime and improves classification performance.

Categories and Subject Descriptors

I.2.6 [**Artificial Intelligence**]: Learning – *concept learning, induction, knowledge acquisition, parameter learning.*

General Terms

Algorithms, performance, experimentation.

Keywords

Classifier systems, data mining, machine learning, optimization, parameter tuning, genetic algorithms, support vector machines.

INTRODUCTION

1.1 Background

The rising rate of medical expenditures warrants effective and efficient healthcare clinical and administrative decisions [1]. These decisions are enhanced by selecting important features that are associated with length of stay, a critical indicator of patient care and clinical and financial outcomes. A typical national healthcare database consists of millions of records and each record consists of hundreds of features that profile patients, hospitals, procedures and other factors. Feature selection allows providers to focus on factors that are most relevant for achieving effective healthcare outcomes.

Support Vector Machine (SVM) is a robust supervised learning machine model that solves linearly non-separable classification problems by mapping an input space into a high-dimensional feature space using a kernel function [2, 3]. Genetic Algorithm (GA) is a class of algorithms that mimics Nature's evolution strategy to evolve a population of chromosomes as potential solutions to optimization problems such as feature selection [4].

This study presents an efficient approach for reducing runtime costs of a SVM-GA hybrid feature selection system for large databases without sacrificing classification performance. Our approach utilizes GA to select relevant features and SVM classifiers to evaluate the importance of those features.

1.2 Problem Statement

A traditional SVM-GA feature selection system, while robust, is computationally infeasible with large datasets. A GA run involves many generations; each GA generation requires numerous SVM model selections for chromosome fitness evaluation; each SVM model selection searches numerous parameters; each parameter set requires a SVM training with runtime complexity of $O(n^3)$. Based on simulation projections with a dataset containing about 3,000 records, this combined complexity requires almost 100 years to complete the feature selection process using a 2.4 GHz Core 2 Duo processor.

1.2.1 Computational Cost of SVM Trainings

A single SVM training has a runtime cost of $O(n^3)$ where n is the size of the dataset [2]. A typical national healthcare database consists of millions of patient records per year. Even for a specific classification problem, such as classifying diabetes patients based on the length of stay at the hospital, the number of records of interest could still be in the thousands per year for a specific age group alone. The sheer size of these datasets is a major contributor to the runtime complexity problem.

1.2.2 Parameter Search for SVM Model Selection

A SVM model selection requires an expensive search for a set of optimal SVM training parameters. For example, a SVM with a radial basis function (RBF) kernel requires the optimization of two crucial parameters, C and γ, where C is a penalty parameter

GECCO'08, July 12–16, 2008, Atlanta, Georgia, USA.
Copyright 2008 ACM 978-1-60558-130-9/08/07...$5.00.

for classification errors and γ affects the width of the Gaussian functions of the RBF kernel. The parameter search involves about 366 unique C-γ pairs or SVM trainings [5]. Also, classifier trainings are usually repeated k times using a k-fold cross validation (CV) process to reduce the bias. If the parameter search requires p SVM training runs with a k-fold CV, the runtime overhead is increased to $O(p \cdot k \cdot n^3)$ where the $p \cdot k$ constant term is rather large.

1.2.3 Combined Computational Cost of a Traditional SVM-GA System

The runtime cost is multiplied over the entire population and evolution process. Assume the following: 1) GA maintains c chromosomes and runs for g generations; 2) the SVM model selection step for fitness evaluation involves a parameter search that requires p SVM trainings; 3) each training has a runtime complexity of $O(n^3)$ for n samples; and 4) a k-fold CV process is used in a SVM training. The overall runtime complexity would be $O(c \cdot g \cdot p \cdot k \cdot n^3)$ with a large constant term $c \cdot g \cdot p \cdot k$. The high runtime time complexity renders a traditional SVM-GA system computationally infeasible.

1.3 Outline of the New Approach

We introduce an innovative combination of techniques to overcome the runtime challenges. A major bottleneck of runtime cost for traditional SVM-GA systems is the SVM training complexity of $O(n^3)$ because SVM trainings are repeated in every step of the evolutionary process. First, we partition the original large dataset into smaller clusters of size $m << n$, effectively reducing the SVM training time for a cluster. After the dataset is partitioned, individual SVM classifiers are trained for each cluster. The trainings are processed in parallel; shortening the overall processing time for the entire dataset. This partitioning strategy is very scalable to increases in dataset size because more clusters can be created for larger datasets.

Second, we employ an adaptive multi-resolution parameter search based on *Uniform Design* [6] that reduces the time to search for optimal parameters. For example, the number of SVM trainings for optimizing the C-γ parameters can be reduced more than 12 times, from 366 to just 30.

Finally, we use caching to reduce the number of redundant trainings in a GA run. GA generates identical chromosomes from time to time, especially when the population is nearly convergent. The proposed caching mechanism eliminates about 30% of the SVM trainings in the experiments.

Details of the new approach are presented in Section 3.

1.4 Previous Work

As a comparison, many traditional SVM-GA hybrid approaches address the SVM training complexity of $O(n^3)$ in *ad-hoc* ways. For example, in [7], the original dataset for text-mining is comparable in size to the healthcare datasets. However, the authors manually select a subset of features and reduce their dataset from over 800,000 samples to about 7,000. Other prior SVM-GA approaches deal with very small datasets with as few as 42 samples [8, 9, 10, 11] and therefore, they do not face the challenge of high runtime complexity caused by large datasets.

For the SVM runtime training parameters, some prior SVM-GA approaches arbitrarily choose only one set of parameters [8, 12] but such arbitrary choice of parameters may not be optimal. In [13], a greedy gradient descent method is proposed to search for an optimal C-γ pair and it assumes that the C-γ space is concave but this is definitely not the case for most applications. Other SVM-GA approaches perform exhaustive parameter searches because their datasets are small [9, 10, 11]. GA is employed to optimize SVM parameters in [14] by encoding those parameters as chromosomes but feature selection is not part of the system.

Finally, our approach not only uses the Least Recently Used caching strategy in [15] but also employ dirty bits to update the chromosome cache.

The rest of the paper is organized as follows. In section 2, the technical background of SVM, GA, and a traditional SVM-GA hybrid system are introduced. Section 3 covers the details of the new approach. Experimental results are presented in Section 4. Finally, conclusion and future work are discussed in Section 5.

2. Technical Background

2.1 Support Vector Machine (SVM)

Support Vector Machine (SVM) is a robust learning machine model that has a wide range of applications in classification and regression problems [2, 3]. This section also introduces two important SVM training parameters C and γ, which are used in the experiments to demonstrate the parameter search for SVM model selection.

Consider a classification problem with a dataset consisting of n instance-label pairs, $S = \{(x_i, y_i)\}$ where $i = 1,\ldots, n$, $x_i \in R^N$ is an instance vector and $y_i \in \{-1,+1\}$ is a class label. Classifier training is essentially a process that finds a hyperplane that separates the positive (+1) samples from the negative (−1) samples. The training process involves the optimization of the following expression:

$$\frac{1}{2} w^T w + C \sum_{i=1}^{n} \xi_i \tag{1}$$

subject to the constraints:

$$y_i (w^T \Phi(x_i) + b) \geq 1 - \xi_i, \; i = 1,\ldots, n, \text{ and} \tag{2}$$

$$\xi_i \geq 0, \; i = 1,\ldots, n \tag{3}$$

where w is the normal vector of the hyperplane, ξ_i, i=1,…,n are the slack variables for measuring classification errors, C is a positive constant or penalty parameter for the error term $\sum_{i=1}^{n} \xi_i$, and Φ is a function that maps the input space to a higher dimensional feature space [3, 16]. The transformation of space is actually a transformation of a linearly non-separable problem to an easier linearly-separable problem in higher dimensions. SVM relies on a kernel function for the transformation:

$$K(x_i , x_j) \equiv \Phi(x_i)^T \cdot \Phi (x_j). \qquad (4)$$

In this study, the *radial basis function* (RBF) kernel is used because it is robust and effective for a wide range of applications. The RBF kernel is defined as

$$K(x_i , x_j) = exp(-\gamma \|x_i - x_j\|^2), \quad \gamma > 0 \qquad (5)$$

where γ is a constant for adjusting the width of Gaussian functions of the kernel.

The constants C in Equation (1) and γ in Equation (5) are two important parameters for SVM model selection and are being used in this study to demonstrate the efficacy of the adaptive multi-resolution parameter search. An optimal setting of C and γ ensures an optimal SVM classifier model. However, finding an optimal C-γ pair involves a computationally expensive grid search on the C-γ plane [5].

For example, consider Figure 1, which is generated from one of our experiments. The classification accuracies of SVMs trained with different C-γ parameters are plotted against $log_2(C)$ and $log_2(\gamma)$. The C-γ search space is non-convex with multiple local optima. The accuracy rates vary greatly from about 52% to 74% and hence a thorough parameter search is essential for finding an optimal classifier model. Prior approaches that arbitrarily select one point on the grid [8, 12] or employ a gradient descent search [13] do not work in general.

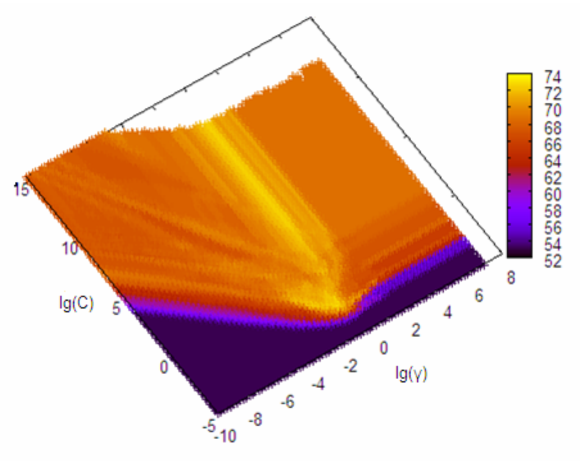


Figure 1: Accuracy of a SVM Plotted on a *C*-*γ* Grid

2.2 Genetic Algorithm (GA)

Genetic Algorithm (GA) maintains and evolves a population of chromosomes as potential solutions to an optimization problem. A new population of chromosomes is reproduced by applying genetic operators such as mutation and crossover on the parent chromosomes in a reproduction step. In the fitness evaluation step, the new chromosomes are assigned fitness values based on an objective function. After that, the population undergoes a "natural" selection process that selects the fittest individuals to mate and reproduce. The above steps are repeated until a specified number of generations is reached. GA has been applied effectively to solve a wide spectrum of optimization problems, including feature selection problems in bioinformatics or biomedical areas [8, 9, 10, 12] and in data mining [7].

2.3 A Traditional SVM-GA Hybrid System

A typical setup of a SVM-GA hybrid system to perform feature selection is depicted in Figure 2. The GA part of the system is responsible for evolving chromosomes as sets of selected features that are important for classification outcomes. In the fitness evaluation step, the classification performance of a SVM classifier model is assigned as the fitness of a chromosome using the following steps:

1. The selected features are used to reduce the dimension of the data by removing unimportant features from the dataset.
2. The reduced dataset is used as training data for a SVM model selection process.
3. SVM model selection involves a parameter search that trains multiple SVMs using different sets of parameters.
4. The best SVM classifier model is selected based on classification performance, which is assigned as the fitness value of the corresponding chromosome.

For a chromosome to survive, it must select relevant features that are essential to SVM classification performance. Hence, at the end of the evolution, the most important set of features are selected.

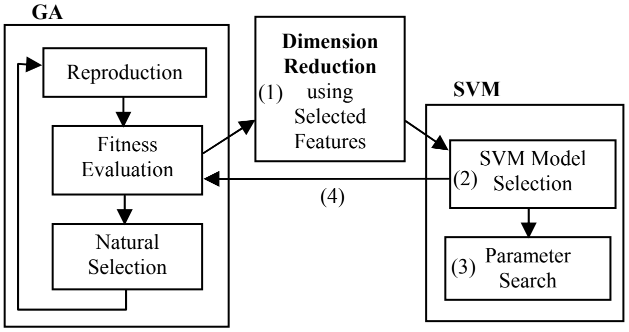


Figure 2: A Traditional SVM-GA Hybrid System

3. Methodology

This study employs a deliberate and novel combination of techniques to reduce the runtime costs of implementing a SVM-GA hybrid system for feature selection. The computational challenges are met with techniques discussed in this section.

3.1 Hierarchical Clustering for Reducing SVM Training Time

Since the training time of SVM is in the order of $O(n^3)$, the most effective way to reduce that complexity is to reduce n, the size of the dataset. Our approach is to apply hierarchical clustering [17] on the original dataset to partition the dataset into smaller clusters based on specified features such as procedure codes. If the size of a cluster m is much smaller than n, the SVM training time for each cluster will be much shorter. Moreover, trainings for the clusters can be done in parallel to shorten the processing time for the entire dataset. Clustering is essential to the scalability of this approach. Larger datasets may simply be partitioned into more clusters that are processed in parallel. The turnaround time can be kept at $O(m^3)$ if there are enough processors available.

3.2 Multi-resolution Parameter Search for SVM Model Selection

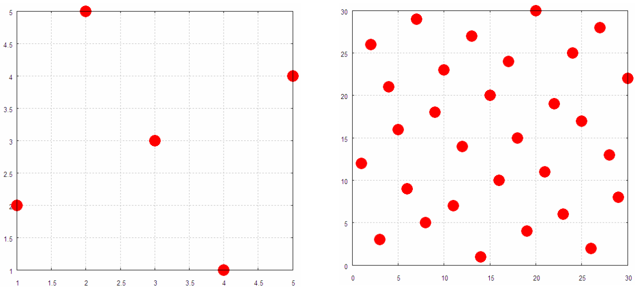
An experiment design technique called *Uniform Design* (UD) is proposed to reduce the number of SVM trainings required in a C-γ search [18]. Uniform Design was originally proposed to reduce the number of experiments required to maintain minimal discrepancy in experiment results [6]. The main goal is to obtain good results without conducting a large number of experiments.

Given a s-dimensional domain, U^s, and a set of m sampling points $P_m = \{\theta_1, \ldots, \theta_1\} \subset U^s$. The points in P_m are uniformly scattered on U^s such that the following L_2-discrepancy of non-uniformity of P_m is minimized:

$$D_2(U^s, P_m) = \left[\int_{U^s} \left| F_m(\theta) - F(\theta) \right|^2 d\theta \right]^{1/2}, \qquad (6)$$

where $F(\theta)$ is the cumulative distribution function over U^s and $F_m(\theta)$ is the empirical cumulative distribution function of P_m. Intuitively, the goal is to spread the sampling points over the experiment domain as uniformly as possible. Figure 3 illustrates two sets of sampling points on a 2-dimensional domain using 5 and 30 sampling points, respectively. Minimization of Equation (6) is a *NP*-Complete problem; luckily, the sampling points can be pre-computed ahead of time and used in different kinds of experiments. Templates of pre-computed points may be downloaded from:

`http://www.math.hkbu.edu.hk/UniformDesign`



(a) 5-Point UD

(b) 30-Point UD

Figure 3: Example of Uniform Designs

Preliminary experiments were set up to employ the 30-point UD on the C-γ grid to reduce SVM trainings in a SVM-GA system, with each point representing a C-γ pair. However, the models obtained from 30-point grid searches are still not optimal because the resolution is too coarse. Since it takes 2 to 3 days to process one cluster with 30 points, increasing the number of UD points is not an attractive option either. Hence, we employ a new adaptive multi-resolution strategy using a global grid and several local grids.

First, during a GA run, a complete 30-point global grid search is performed on the entire C-γ plane every 10 generations. Each C-γ point corresponds to a SVM model. The top 6 distinct points from the 30-point global search are selected based on the classification performance of the SVM models averaged over the population. Subsequently, each of the top 6 points is used as the center of a 2-unit×2-unit local grid using the 5-point UD template. These 6 local grids are used in the next 9 generations for selecting the best SVM model for each chromosome. With 6 local grids and 5 points per grid, a total of 30 points are still being searched per generation but the searches are now capable of focusing on promising local regions on the C-γ plane. This is an adaptive multi-resolution grid search because the search alternates between a global scale and a local scale. The resolution is reset to the global 30-point grid every 10 generations to adapt to changes in the global C-γ landscape over time. By doing periodic global searches, global changes can be detected and yet the local grids still enable the search to zero-in to the local optimal points.

Using this method, the total number of SVM trainings per chromosome per generation is reduced from 1,464 (4×366 for a 4-fold Cross Validation) to 120. Suppose the population size is maintained at 200 and the number of generations is set at 250. The adaptive multi-resolution search reduces the total number of SVM trainings from 73.2 million to 6 million for each GA run.

3.3 Reducing Computational Cost of GA by Caching

As GA generally converges over time, the chromosome population exhibits more occurrences of identical chromosomes towards the end of the evolution. To further speed up the evolutionary process, a caching system is employed to avoid redundant SVM trainings. A chromosome and its optimal SVM model for a particular C-γ are stored in the cache. While evaluating the fitness of a new chromosome, if a copy of the same chromosome is found in the cache, the fitness value of an existing SVM model will be used directly without any further training. In addition, a dirty bit is associated with each chromosome. If a chromosome is reproduced as a clone of its parent, the parent's fitness and SVM model will be directly inherited without any cache search. The cache is also updated continuously to remove the least frequently used entries. This turns out to be a very effective cost saving strategy. As shown in the experimental results section, the average cache hit rate is about 30%, reducing the number of trainings by almost a third.

3.4 Fitness Function for Feature Selection

Accuracy is used as the fitness measure in many traditional SVM-GA hybrid approaches [8, 9, 12]. However, accuracy alone is not measuring other important factors such as true positive rates. This study utilizes *Accuracy*, *Recall*, *Precision*, and *F-Measures* for performance measurements [19].

Let *TP* (True Positive) be the number of positive samples that are classified correctly; *FP* (False Positive) be the number of negative samples that are classified as positive incorrectly; *TN* (True Negative) be the number of negative samples that are classified correctly; and *FN* (False Negative) be the number of positive samples that are classified as negative incorrectly. Hence, *Accuracy* of a classifier is defined as

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN}. \qquad (7)$$

Accuracy measures the percentage of samples that are classified correctly. *Precision* measures how many of the positively classified samples are indeed positive, and is defined as:

$$Precision = \frac{TP}{TP + FP}. \qquad (8)$$

Recall measures how many of the actual positive samples are classified correctly as positive. *Recall* is defined as:

$$Recall = \frac{TP}{TP + FN}. \qquad (9)$$

Precision and Recall can be combined into a single term called *F-Measure* where

$$F\text{-}Measure = \frac{2 \times Precision \times Recall}{(Precision + Recall)}. \qquad (10)$$

F-Measure is essentially a harmonic mean of *Precision* and *Recall*. The *F-Measure* enforces a trade-off between *Precision* and *Recall* to ensure a good balance between false positives (*FP*) and false negatives (*FN*). Finally, the fitness function ψ for chromosome fitness evaluation balances Accuracy, Precision, and Recall and is defined as

$$\psi = (Accuracy + F\text{-}Measure) / 2 \qquad (11)$$

4. Experimental Results

4.1 Dataset

This study utilizes the Healthcare Cost & Utilization Project (HCUP-3) database, which is the largest and most robust U.S. national inpatient database [20]. An outpatient record in HCUP consists of more than a thousand variables including patient demographic data, hospital profile, patient diagnoses and procedures, total hospital charges, and length of stay. The SVM-GA systems in the experiments are designed to use only treatment procedures to classify patient records with respect to length of stay. Based on a total of 231 possible procedure codes, records of type-2 diabetes patients with at least two procedure codes for the year 2004 are included in the experiments. The dataset is further restricted to include records for patients 65 years or older and on Medicare. The coding for patients on Medicare is more accurate due to federal reimbursement guidelines [21]. Data cleaning is also performed to remove records with incomplete or invalid information. The resulting dataset consists of a total of 3,115 records; although larger datasets will scale well with more clusters being processed in parallel. As noted earlier, even without using GA, regular SVM trainings on such a dataset are expensive because of the $O(n^3)$ complexity. Furthermore, this is a two-class classification problem. Samples with length of stay less than 9 days, the median length of stay for the dataset, are labeled as positive samples; otherwise, they are labeled as negative samples.

4.2 Experiment Setup

The original dataset of 3,115 samples is first clustered into 11 clusters based on procedure codes using hierarchical clustering. The sizes of the clusters are as shown in Table 1. The average cluster size is 283.18.

Table 1: Summary of Cluster Sizes

Cluster ID	No. of Samples
0	258
1	330
2	299
3	280
4	308
5	267
6	263
7	258
8	319
9	264
10	269
Average	283.18

The GA experiments are repeated for each of the clusters. The chromosome population size is set at 200, the maximum number of generations is set at 250, the crossover rate is 0.6 and the mutation rate is 0.01. The natural selection process involves a tournament selection [22] that selects three-fourths of the next population. The remaining one-fourth of the population is selected by a roulette wheel selection [4]. Each GA experiment is repeated five times per cluster and the results are tallied and averaged. In this study, an average performance measure is calculated as a weighted average over all clusters based on the number of samples in each cluster:

$$weighted\_average = \frac{1}{n}\sum_i p_i \cdot m_i, \qquad (12)$$

where p_i is a performance measure and m_i is the number of samples in Cluster *i*, respectively, and *n* is the total number of samples in the original dataset. The weighted average ensures fair and proportional contributions from clusters of different sizes. The SVM Light software package [5] is used for SVM modeling selection.

4.3 Feature Selection Results

The average numbers and percentages of features selected are listed in Table 2. The results are based on the average number of features of the best chromosomes over 5 GA runs. Out of a total of 231 procedure codes, the overall weighted average number of selected features is 27.78 (or 12.02%). The percentages of features selected for the clusters vary between 8% and 17%. The results support our initial conjecture that only certain features are associated with length of stay.

4.4 Comparisons of Classification Performance

To confirm that the features are selected because of their importance for classification performance, a control experiment is set up to verify that the new approach selects relevant features while maintaining classification performance. Classification performance of the new approach is defined as the weighted average of the performance of each cluster's SVM classifier model.

Table 2: Feature Selection Results

Cluster ID	# of Selected Features	% of Selected Features
0	25.43	11.01%
1	39.20	16.97%
2	19.00	8.23%
3	24.40	10.56%
4	26.56	11.50%
5	19.20	8.31%
6	22.80	9.87%
7	32.20	13.94%
8	35.00	15.15%
9	35.50	15.37%
10	23.67	10.25%
Weighted Average	**27.78**	**12.02%**

The control experiment uses a single classifier to train on the entire dataset without feature selection to set up a baseline expectation for classification performance. The optimal classifier model in the control experiment is selected using a two-stage 366-point grid search based on the findings of Hsu, Chang, and Lin [5]. The weighted averages of the fitness value, *Accuracy*, *Precision*, and *Recall* of the new approach and the control experiment are summarized as percentages in Figure 4.

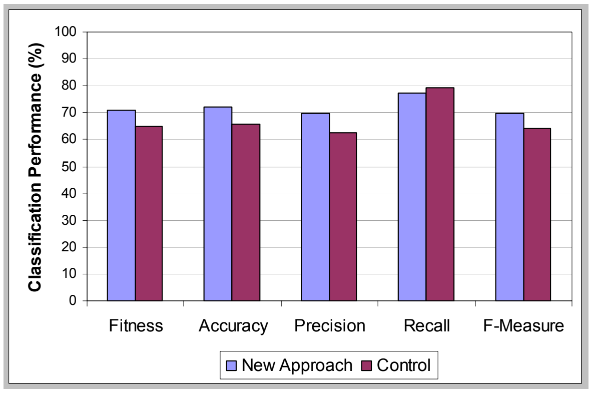


Figure 4: Classification Performance Comparisons of New Approach vs. Control

The overall classification performance of the new approach outperforms the control even when a small subset of features is selected. The fitness value ψ, *Accuracy*, *Precision*, and *F-Measure* of the new approach exceed those of the control system by 6.09, 6.59, 7.17 and 5.67 percentage points, respectively. Although the new approach shows 2.1 percentage points less on *Recall*, the *F-Measure* indicates that the new system has a better balance of *Precision* and *Recall*. The new approach not only maintains classification performance, it actually exceeds the classification performance of the control. This may be attributed to the fact that the unimportant features are noise; once the noise is removed from the dataset, SVMs trainings become more effective. Therefore, the new approach is successful in selecting relevant features that are most important for classification.

4.5 Adaptive Multi-Resolution Search

To illustrate the adaptive multi-resolution search of the new approach, the local grids from an all-time best chromosome for a cluster are overlaid on the cluster's *C*-γ plane, see Figure 5. To generate a more complete global *C*-γ fitness map, additional SVM classifiers are trained over the entire *C*-γ plane after GA terminates. The fitness values of the classifiers are plotted in the background to show where the promising regions are located. A lighter color indicates a higher fitness region whereas a darker color indicates a lower fitness region. The local grids of the all-time best chromosomes from one of the GA runs for Cluster 3 are displayed as black dots. Figure 5 demonstrates that the local grids of the multi-resolution search migrate to the promising regions on the *C*-γ plane.

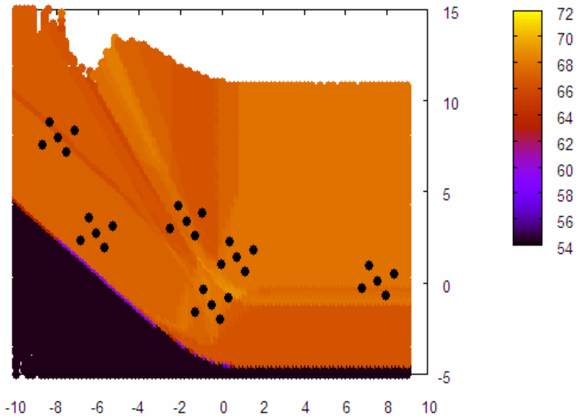


Figure 5: Adaptive Multi-Resolution Search on the *C*-γ Plane

4.6 Comparison of Multi-Resolution Search and Arbitrary Parameter Selection

Some previous SVM-GA approaches [8, 12] use a pre-selected set of parameters in SVM trainings. To demonstrate that arbitrary selections of SVM training parameters are insufficient to produces optimal performance, a second control experiment is set up to replicate the new approach. This experiment uses a pre-selected *C*-γ point for SVM model selection instead of using the multi-resolution *C*-γ search.

To pre-select a *C*-γ point for each cluster, an arbitrarily large number (1,100) of SVM classifiers are first trained corresponding to 1,100 points on the *C*-γ plane. A specific *C*-γ point is chosen by selecting the best SVM classifier model out of 1,100 SVM classifiers. The pre-selected *C*-γ point is then used in the control system to train SVMs throughout the evolutionary process. The weighted averages of fitness, *Accuracy*, *Precision*, and *Recall* for the new approach and the control are charted in Figure 6. Clearly, the performance of the control system is inferior to the new approach across the board despite the laborious effort to pre-select *C*-γ. Using an arbitrarily fixed set of parameters to train SVM is really not appropriate because it is essentially selecting an *a priori* SVM model without examining the changing solution space during the evolutionary feature selection process.

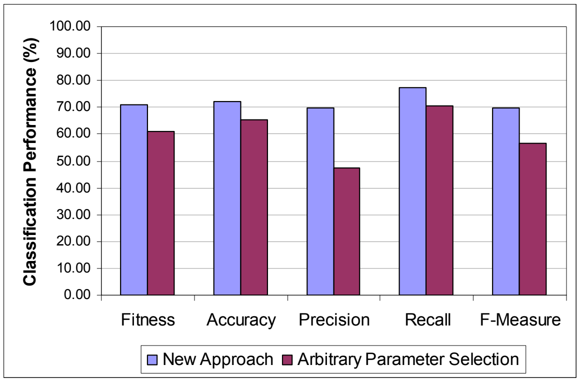


Figure 6: Classification Performance Comparison of Multi-Resolution Parameter Search vs. Arbitrary Parameter Selection

4.7 Chromosome Caching

The average cache hit rates are listed in Table 3. The cache hit rates are calculated as the total number of fitness evaluations cached divided by the total number of fitness evaluation required. An actual fitness evaluation is cached if a chromosome is found in the cache or if it is a direct clone of its parent. The weighted average of cache hit rate is 30.17% over all clusters. The caching algorithm reduces the runtime of the system by almost a third. Some clusters achieved high hit rates, such as 54.50% and 44.71% for Clusters 2 and 5, respectively.

Table 3: Cache Hit Rates Summary

Cluster ID	Cache Hit Rate
0	32.91%
1	19.64%
2	54.40%
3	26.98%
4	31.07%
5	44.71%
6	28.06%
7	20.55%
8	24.25%
9	21.49%
10	28.21%
Weighted Average	**30.17%**

4.8 Execution Time

The execution time required for the new approach to process the clusters is listed in Table 4. The maximum execution time for a cluster is 2.38 days (cluster 8) using a Core 2 Duo 2.4 GHz processor. When all 11 clusters are processed in parallel, this 2.38 day maximum execution time is the turnaround time for the new approach.

By comparison, the control system described in Section 4.4 took 17.42 hours to select a single SVM classifier model using the entire dataset of 3,115 samples. If a traditional GA is added to the control system to perform feature selection, it is projected that turnaround time would be 99.4 years (17.42 hours per SVM model selection × 200 chromosomes × 250 generations). As a result, no experiments can feasibly process the entire data set using a traditional SVM-GA hybrid system.

Nevertheless, any large dataset can be partitioned into small enough clusters that can be processed in parallel. The new approach is very scalable and the turnaround time is relatively low given a small cluster.

Table 4: Execution Time for GA Runs

Cluster ID	Execution Time (Days)
0	1.61
1	1.84
2	1.55
3	1.62
4	1.72
5	1.48
6	1.63
7	1.43
8	2.38
9	2.22
10	1.86

5. Conclusion and Future Work

This paper presents an efficient and effective SVM-GA hybrid model to perform feature selection for large datasets. The new approach successfully selected important features for a healthcare dataset involving 3,115 hospital discharge records and 231 features. This approach is scalable for any large dataset.

The major bottleneck in a traditional SVM-GA system is caused by the size of the dataset because the SVM training complexity is $O(n^3)$. As long as the size of each cluster is reasonably small, the proposed techniques of using an adaptive multi-resolution Uniform Design to perform parameter search and chromosome caching can greatly reduce the runtime costs of GA and yet maintain strong classification performance. A very large dataset could be partitioned into many small clusters of size m. As long as $m << n$ and parallel machines are available to process the clusters, the system performance will scale well with the data size.

Future study will extend this framework using more features such as diagnosis, patient profiles, and hospital profiles in order to improve feature selection performance. Features selected in the experiment will be evaluated for their clinical significance and implications. This project shows promise for developing decision support systems that will assist healthcare professionals in making better clinical and administrative decisions.

6. Acknowledgments

This research was supported in part by the Student Research Assistant Program and Research Incentive Award from the University of South Carolina Upstate.

7. REFERENCES

[1] DeVol, R. and Bedroussian, A. 2007. An unhealthy america: the economic burden of chronic disease, Miliken Institute, Santa Monica, California.

[2] Vapnik, V. 1998. Statistical learning theory, John Wiley&Sons, Inc., New York.

[3] Burgess, C. 1998. A tutorial on support vector machines for pattern recognition. Data Mining and Knowledge Discovery, 2, Kluwer Academic Pub., Boston, 121-167.

[4] Goldberg, D.E. 1989. Genetic algorithms in search, optimization and machine learning, Kluwer Academic Publishers, Boston, MA.

[5] Hsu, C.W., Chang, C.C., and Lin, C.J. 2007. A practical guide to support vector classification. Technical report, Dept. of Comp. Sci. & Info. Engr., National Taiwan University.

[6] Fang, K.T., Shiu, W.C., and Pan, J.X. 1999. Uniform designs based on Latin squares. Statistica Sinca, 9, 905-912.

[7] Morariu D., Vintan L. Tresp V. 2006. Evolutionary feature selection for text documents using the svm. Proceedings of the 3rd International Conference on Neural Networks and Pattern Recognition, NNPR06, Barcelona, October, 2006.

[8] Huerta E.B., Deval, B., and Hao, J. 2006. A hybrid GA/SVM approach for gene Selection and classification of microarray data. EvoWorkshop 2006, LNCS 3907, 34-44.

[9] Li, L., Jiang, W., Li X., Moser, K.L., Guo, Z., Du, Lei, Wang, Q., Topol, E.J., Wang Q., and Rao, S. 2005. A robust hybrid between genetic algorithm and support vector machine for extracting an optimal feature gene subset. Genomics 85, Elsevier, 16-23.

[10] Liu J.J., Cutler, G., Li, W., Pan, Z., Peng, S., Hoey, T., Chen, L., and Ling X.B., Multiclass cancer classification and biomarker discovery using GA-based algorithms. J. Bioinformatics, 21(11), 2691-2697.

[11] Zhao, X., Huang, De, Cheung, Y., Wang, H., and Huang X. 2004. A novel hybrid GA/SVM system for protein sequences classification. IDEAL 2004, LNCS, 3177, 11-16.

[12] Agrawal, R.K. and Bala, R. 2007. A hybrid approach for selection of relevant features for microarray datasets. Intl. J. Computer and Information Science and Engineering, 1(4), 196-202.

[13] Bao, Y. and Liu, Z. 2006. A fast grid search method in support vector regression forecasting time series, LNCS, 4224, 504-511.

[14] Lessmann, S., Stahlbock, R., Crone, S. 2005. Optimizing hyperparameters of support vector machines by genetic algorithms. Proceedings of the International Conference on Artificial Intelligence, ICAI'05, Las Vegas, CSREA Press: Athens, Vol. 1, pp. 74-80.

[15] Kratica, J. 1999. Improving Performances of the Genetic Algorithm by Caching. Computers and Artificial Intelligence, 18(3), 271-283.

[16] Cortes, C. and Vapnik V. 1995. Support-vector network. Machine Learning, 20 (Sep. 1995), 273-297.

[17] Jain, A.K., Murty M. N. and Flynn P. J. 1999. Data clustering: a review. ACM Computing Surveys, 31, 264-323.

[18] Huang, C. & Lee, Y., Lin, D., and Huang, S. 2007. Model selection for support vector machines via uniform design. Comput. Stat. Data An., 52(1), (Sep. 2007), Elsevier, 335-346.

[19] Singhal, A. 2001. Modern information retrieval: a brief overview. Bulletin of the IEEE Computer Society Technical Committee on Data Engineering 24 (4), 35-43.

[20] Dowell, M., Rozell, B., Roth, D., Delugach, H., Chaloux, P. and Dowell, J. 2004. Economic and Clinical Disparities in Hospitalized Patients with Type-2 Diabetes. Journal of Nursing Scholarship, 36, 66–72.

[21] Kiyota, Y., Schneeweiss, S., Glynn, R., Cannuscio, C., Avorn, R., & Solomon, D. 2004. Accuracy of medicare claims-based diagnosis of acute myocardial infarction: estimating positive predictive value on the basis of review of hospital records. American Health Journal, 148(1), 99-104.

[22] Miller, B.L. and Goldberg, D.E. 1995 Genetic algorithms, tournament selection, and the effects of noise. Complex Systems (June 1995), 193-212.