

Using Singular Value Decomposition to Improve a Genetic Algorithm's Performance

Jacob G. Martin
Computer Science
University of Georgia
Athens, GA 30602
martin@cs.uga.edu

Khaled Rasheed
Computer Science
University of Georgia
Athens, GA 30602
khaled@cs.uga.edu

Abstract- The focus of this work is to investigate the effects of applying the singular value decomposition (SVD), a linear algebra technique, to the domain of Genetic Algorithms. Empirical evidence, concerning document comparison, suggests that the SVD can be used to model information in such a way that provides both a saving in storage space and an improvement in information retrieval. It will be shown that these beneficial properties can be extended to many other different types of comparison as well. Briefly, vectors representing the genes of individuals are projected into a new low-dimensional space, obtained by the singular value decomposition of a gene-individual matrix. The information about what it means to be a good or bad individual serves as a basis for qualifying candidate individuals for reinsertion into the next generation. Positive results from different approaches of this application are presented and evaluated. In addition, several possible alternative techniques are proposed and considered.

1 Introduction

The technique of singular value decomposition (SVD) has proven itself valuable in several problem domains. Some examples include data compression[1], image recognition and classification[2], chemical reaction analysis[3], and document comparison[4]. Although these domains are quite different in some aspects, each can be reduced to the problem of ascertaining or ranking relevance in data. Intuitively, the concept of relevance depends critically on the nature of the problem at hand. The SVD provides a method for mathematically discovering correlations within data. Latent semantic indexing (LSI) is a technique of using the SVD to discover and expose latent, or hidden, relationships between terms in documents. Although similar vector-based models exist for analyzing relationships within documents, such as the QR factorization, empirical evidence concerning document comparison suggests that LSI actually improves the quality of the information retrieved[5].

The focus of this paper is to investigate the impact of the SVD on Genetic Algorithms (GAs). The SVD is used to

expose the most striking similarities between a given individual and a strategically chosen population of individuals. These similarities are used to influence the direction of the GA's search process by qualifying candidate individuals for reinsertion into the next generation based on their proximity to other individuals, whose fitnesses have already been computed. Initial results from the application of this process indicate significant improvements in the GA's performance. The intent is to evaluate several different tested approaches of using SVD qualifiers to enhance the performance of GAs, justify any apparent performance improvement over traditional GAs, and to speculate about other ways of using SVD in GAs. In the remainder of this paper we provide some background information in Section 2, followed by a description of the proposed method for SVD integration into a genetic algorithm in Section 3. Section 4 describes the results achieved by using the proposed method in several different problem domains. Finally, Section 5 provides some promising opportunities for future research.

2 Background

Let A be a $p \times q$ matrix of rank r whose rows represent genes and columns represent individuals. The singular value decomposition expresses A as the product of three matrices.

$$A = UDV^T \quad (1)$$

Where U is a $p \times p$ matrix whose columns are made up of the orthonormal eigenvectors that span the space corresponding to the gene-gene auto-correlation matrix AA^T . Likewise, V is a $q \times q$ matrix whose columns consist of the orthonormal eigenvectors that span the space corresponding to the individual-individual auto-correlation matrix $A^T A$. The middle matrix, D , is a $p \times q$ diagonal matrix with $D_{ij} = 0$ for $i \neq j$ and $D_{ii} = S_i \geq 0$ for $\forall i$. The S_i 's are arranged in descending order with $S_1 \geq S_2 \geq \dots \geq S_n$, where $n = \text{Min} \{p, q\}$. The S_i 's are called the singular values of A (the square roots of the eigenvalues of AA^T and $A^T A$). For a formal proof of the SVD's existence, see [3]. The singular values indicate the weight, or importance, of a dimension. LSI works by setting all but the k largest

singular values in the decomposition to zero. This results in the formation of a new low-dimensional matrix A_k , of rank k , corresponding to the k most influential dimensions. Intuitively, this new subspace exposes the most important similarities and suppresses the insignificant correlations between the vectors represented in the matrix A . Exactly how much of the original space is preserved is directly related to the amount of reduction performed. A theorem by Eckart and Young states, informally, that the new low-dimensional matrix obtained is the closest matrix among all matrices of its rank to the original matrix [6]. Formally, it states that among all $p \times q$ matrices C with rank at most k , A_k is the one that minimizes

$$\|A - C\|_F^2 = \sum_{i,j} (A_{ij} - C_{ij})^2 \quad (2)$$

Although this may explain why the reduction doesn't "deteriorate too much in performance over conventional vector-space methods; it fails to justify the observed improvement in precision and recall" [7]. However, several papers have made positive steps towards a rigorous proof that, given an appropriate representation, the benefit is achieved with high probability [7][8].

3 Incorporation Into a Genetic Algorithm

Genetic Algorithms are search and optimization methods that mimic natural selection and biological evolution to solve problems. For a thorough introduction to Genetic Algorithms (GAs) please see the books by Goldberg[9] and Michalewicz[10]. How can SVD be incorporated into a genetic algorithm? There are several possible approaches, but the one considered and evaluated here is based on qualifying candidate individuals for reinsertion. Reduced models of a GA's optimization history have been used in previous works to help direct the search process[11]. The idea of qualifying individuals, before evaluating their fitness, and reinserting them into the next generation has been visited in [12]. The idea there was to use the two nearest neighbors of the potential individual to qualify it. For example, for smooth functions like the sphere model, a linear interpolation between similar individuals should be one of the best approximation approaches one can find.

It has been shown experimentally and probabilistically that the SVD should be able to expose the most striking similarities between a given vector and another set of vectors[7]. These similarities will be used to influence the direction of the GA's search process by qualifying candidate individuals for reinsertion into the next generation based on their proximity to other individuals. One benefit of this approach is that the fitness function need not be computed in order to determine that an individual closely resembles another individual whose fitness is already known. For prob-

lems that require a computationally expensive fitness function, such as those found in engineering design optimization, this benefit could be very significant.

3.1 Qualification

Our qualification approach involves comparing the candidate to the worst individuals in the current population. The qualification process is initialized for each generation by first creating a matrix containing the individuals to qualify candidates against. This matrix is composed of individuals in the current population whose fitnesses are less than half of the current population's average fitness. Conceptually, the subspace spanned by this matrix outlines the qualification guidelines for the current generation. The qualification subspace is then reduced to k dimensions by computing the SVD and eliminating all but the k largest singular values. A readily apparent criticism of the qualification process is that computing the entire SVD at each generation, for large dimensionalities, may become computationally expensive. However, methods exist for both "folding in" new vectors and removing old vectors from an existing SVD computation[13].

Qualification for a candidate individual is based on its proximity to the individuals in the qualification space. In order to compute its proximity, a candidate is first converted into a vector, whose components represent the genes of the individual. The vector is then converted into a unit vector and projected into the qualification space using the diagonal matrix D , which contains the k largest singular values along its diagonal. Assuming a good representation, similar individuals will be represented by nearly parallel vectors and dissimilar individuals by nearly orthogonal vectors. Thus, the concept of similarity is reduced to computing the cosines of the angles between the projected candidate vector and every other vector in the rank- k qualification space. The cosine of an angle is an appropriate function to use because its value approaches one as two vectors become more parallel. Likewise, as two vectors become more perpendicular, the cosine of the angle between them approaches zero. The cosines are then compared to the d -close parameter, which represents the required amount of proximity for qualification ($0 \leq d \leq 1$). If the candidate is at least d -close to any of the worst individuals in the current population, it is discarded. Otherwise, it is allowed a chance to survive to the next generation.

Interestingly, experimental results indicate that if the qualification space is instead composed of the worst individuals over the entire history of the GA, the performance achieved is not nearly as dramatic in comparison with the previously described approach. One reason for this may be that the current state of the system is more accurately captured when comparing similar vectors.

3.2 Transformation

The preceding discussion works under the assumption that similar individuals will be represented by nearly parallel vectors, and dissimilar ones by nearly perpendicular ones. This assumption illustrates the need for a good vector based model for each particular optimization problem. A good model would place good individuals near other good individuals in the individual representation space and likewise for bad individuals. Sometimes, the actual parameters that represent the genes may not lend themselves to accurate categorization of an individual during comparison. For this reason, transformations are applied to each gene of each individual in an attempt to more effectively categorize their values. In continuous function optimization problems, each gene component of an individual is categorized into a section, based on the allowed range of each gene. For example, an individual with ten genes, with gene values varying between zero and ten, could be transformed into an individual with twenty genes. The new genes no longer represent the exact gene values. Rather, they represent the location in a partition of the allowed range of values for the gene. Under this example, the first two genes of the new representation indicate whether or not the individual's original gene before transformation was a high (greater than 5) or low (less than or equal to 5) value. The transformations have the effect of categorizing solutions together based on the interactions between each variable. Here, the SVD is used to expose the successful, or unsuccessful, combinations of high and low values of parameters for a particular function. In comparison to the naive approach (with no transformation), the observed improvement in the qualification process is significant for some problems. However, for other problems, the process was not beneficial. Presumably, this was because the transformation did not accurately represent what it meant to be a good or bad individual for that particular problem.

3.3 Reduction

The amount of reduction performed on the gene-individual matrix, A , directly influences the accuracy achieved. Although the optimal amount of reduction performed is not known a priori for a given problem or situation, there is a technique that may provide a good guess. First of all, if the " k largest singular values of a matrix A are well-separated from the remaining singular values, then the subspace spanned by the corresponding singular vectors is preserved well when a small perturbation is added to A "[7]. The amount of relative change produced by a given reduction to rank k is based on the k largest singular values. It is easily calculated by the following formula (where the sub-

script F denotes the Frobenius Norm).

$$percenterror = \frac{\|A - A_k\|_F}{\|A\|_F} * 100 \quad (3)$$

It is clear that the amount of perturbation strongly influences the percent error. The results presented in the next section are achieved with the policy of reducing to the lowest rank that should causes no more than 10 percent error. Importantly, this isn't a hard-fast rule, and may change depending on problem domain, representation, and stage of evolution. The results produced by varying the error margin (and therefore the rank chosen) at different stages using strategically chosen conditions should be a topic of study for future works.

4 Results

4.1 Overview

Tests were performed using a custom GA, implemented entirely in JavaTM. The source code and documentation for the GA may be obtained by emailing one of the authors. An approach similar to the $(\mu + \lambda)$ evolution strategy was used with populations of size 100 generating 100 candidate individuals. Rank-based selection was used to select individuals for breeding. The breeding process used one point crossover and a ten percent chance of mutation. Mutation was performed by first randomly choosing a gene and then increasing or decreasing it by a random amount between 0 and 1. Reinsertion was achieved by picking the best 100 individuals out of the 200 parents and qualified offspring. To handle situations where not enough children are qualified, the qualifier is disabled after 500 unsuccessful attempts at generating the 100 children. The breeding process then finishes by generating the required number of offspring without qualification. A more ideal solution would be to adjust the amount of reduction and d -close parameters based on some function of the qualification pressure.

All of the results are based on the average of the best individual, for each generation, over 30 different random initial populations. The fitness of an individual is defined as

$$fitness = \frac{1}{1 + |f(x) - target|} \quad (4)$$

The SVDs were computed using Visual Numeric's Java Numerical Library (JNL at www.vni.com). For an in depth explanation of how to compute the rank- k cosines between a query vector and the vectors contained in a reduced rank- k model efficiently see[13]. The GA was tested on a variety of problems, with varying degrees of difficulty and dimensionality, using three different approaches. In the first approach, no rank reduction was performed on the gene-individual matrix at all. The second approach attempted to

determine the best amount of rank reduction automatically by analyzing the separation of the singular values in order to select a rank that would cause no more than 10 percent error. In order to compare the performance of the SVD GA to traditional GAs, the final approach did not incorporate the SVD qualification process at all. These tests indicate that the rank reduction cases, on average, outperformed the plain GA, and the no reduction case.

4.2 Application Domains

The effects of the SVD qualification process were evaluated by testing three different optimization problems. Each problem's goal is to minimize a function. Recall that a higher fitness value corresponds to a lower function value for each particular problem.

4.2.1 N-Queens.

The first problem tested was the n -queens problem. The goal is to find a way to place n queens on an $n \times n$ chess board such that no two queens attack each other. This means that no two queens may be placed on the same row, column, or diagonal. For this reason, the representation chosen was a permutation of the set $\{1, \dots, n\}$. This representation restricts the search to cases in which queens are on separate rows and columns. The function f of an individual x is computed by counting the number of diagonal conflicts in its representation. The fitness of the individual is then computed with equation 4. The results achieved when $n = 30$ are shown in Figure 1. The results indicate that domains with permutation representations are amenable to the SVD qualification process without transformation. We also incur the added bonus that the act of reducing the qualification matrix, thereby bringing out its latent characteristics, outperforms the no-reduction cases. The transformation step was skipped for this problem because the discrete nature of the domain makes it doubtful that significant improvements can be achieved.

4.2.2 Sphere Model Minimization.

The second problem was to minimize the sphere model function, given below.

$$f(x) = \sum_{i=1}^{30} (x_i - 1)^2, x_i \in [-5, 5] \quad (5)$$

The representation chosen for individuals in this problem was a vector of doubles. Transformations were applied by categorizing each of the thirty genes into one of four equal sized partitions of the interval $[-5, 5]$. From Figure 2, it can be seen that the SVD GA outperformed the plain GA once again. However, the act of reduction and transformation did

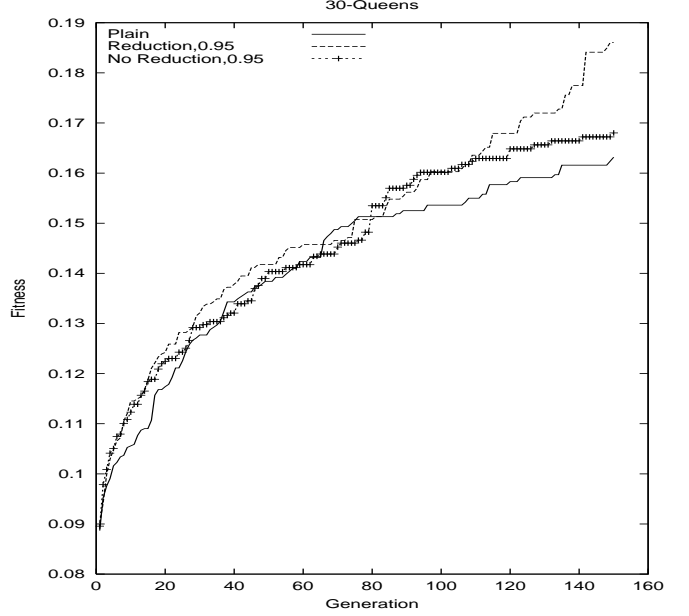


Figure 1: Results for the 30 Queens function

not provide significantly better results over the plain GA in this problem. Presumably this was because the transformation chosen for comparison via the SVD did not adequately capture this problem domain.

4.2.3 Langermann Function Minimization.

The third minimization problem tested was a modified Langermann function.

$$f(x) = - \sum_{i=1}^N c(i) (e^{\frac{-1}{\pi} \|x - A(i)\|_f^2} \cos(\pi \|x - A(i)\|_f^2)) \quad (6)$$

$$x_i \in [0, 10] \quad (7)$$

The GA was tested on the 10 dimensional formulation of this function ($N = 10$). The global minimum for this function is approximately -1.4999. The representation chosen for individuals in this problem was also a vector of doubles. Transformations were applied by categorizing each gene into one of four equal sized partitions of the interval $[0, 10]$. From Figure 3, it is apparent that GAs using the rank- k SVD qualifier outperformed the no reduction cases. In addition, the transformation processes outperformed their counterparts by a significant margin. The SVD process without transformation did not perform as well as the Plain GA. Therefore the type of representation is an extremely important factor for success.

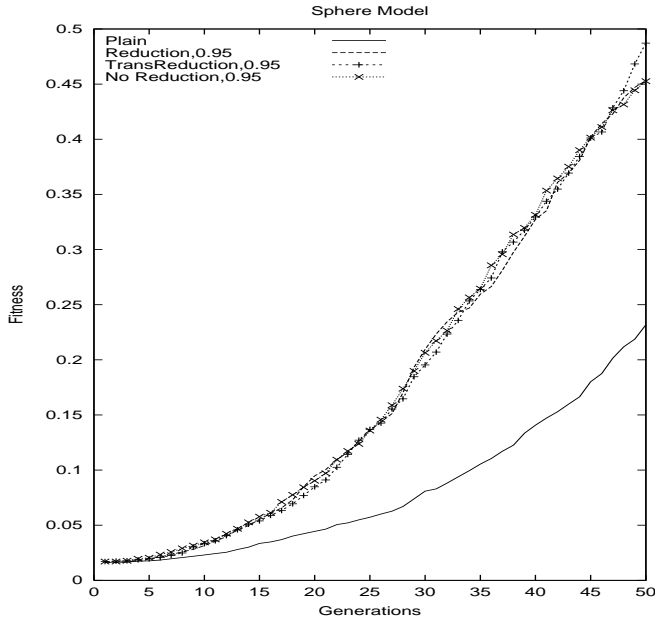


Figure 2: Results for the sphere model

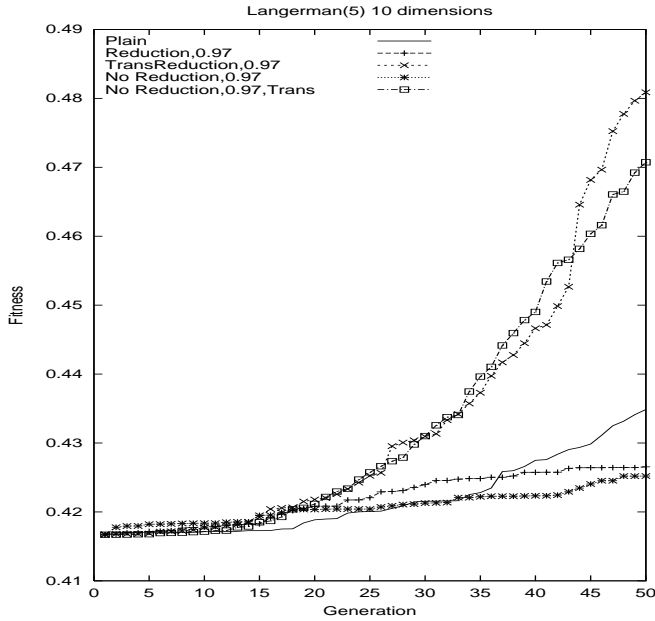


Figure 3: Results for the 10 dimensional Langerman-5 function

5 Future Directions

5.1 Skewness

Using an idea described in [7], we may be able to discover many pieces of important information about a given qualification space. Let Q represent the qualification space. For each individual $q_i \in Q$, let v_q be the vector assigned to the individual by the rank- k SVD on Q . The rank- k SVD is δ -skewed on the qualification space Q if, for each pair of individuals q and q' , $v_q \cdot v_{q'} \leq \delta \|v_q\| \|v_{q'}\|$ if q and q' belong to different categories and $v_q \cdot v_{q'} \geq 1 - \delta \|v_q\| \|v_{q'}\|$ if they belong to the same category. From Papadimitriou et al's findings in [7], the SVD should be able to provide several, probabilistically verified, techniques for problems with appropriate representation. Several pieces of key information about the GA's current state and qualification pressure can be discovered by averaging the values for δ for every pair of vectors between the two qualification spaces Q_g and Q_b . The spaces Q_g and Q_b are composed of good and bad individuals, respectively. First of all, the average of these values describes how well the chosen representation categorizes individuals. Secondly, if the amount of δ -skewness is high then the good aren't far from the bad, and therefore there is more qualification pressure on candidate individuals. This information could be used to indicate when the population has become stale. Furthermore, an algorithm could deduce how skewed the current views of good and bad are, and adjust the amount of the d -close parameter appropriately.

5.2 Gene Clustering

The preceding discussion only makes use of the individual qualification space. Using the eigenvectors that span the gene-gene autocorrelation matrix AA^T , the results could be improved by clustering the different types of good or bad individuals into groups. The information provided from this matrix would be able to show which genes are used similarly across a collection of individuals. From this information, it should be determinable which parts of the problem are decomposable into sub-problems. The GA could then focus its work on optimizing these subproblems, instead of the entire problem.

5.3 Parameter Choices

Unfortunately, it is hard to predict, a priori, the optimal transformation and parameter choices. What is needed is a concrete function with the ability to compute good parameters for a given situation. What makes parameters or representations "good," and under what assumptions for a given problem and situation? The amount of error used should depend strongly on which stage of evolution the GA is in. In the early stages, the GA should be allowed to explore the

search space as widely as possible, in order to find as many basins of attraction as possible. In effect, the GA should avoid having too many similar individuals in the first stages, be they poor or good individuals.

6 Conclusion

In this paper, we presented a method for improving a genetic algorithm's performance by using singular value decomposition to qualify candidate individuals. Results from several application domains show that using the SVD qualifier was significantly beneficial. Furthermore, it was observed that the d -close parameter, the amount of rank reduction, and choice of transformation greatly influence the amount of performance improvement achieved. It is clear that further testing and development on several different types of problems and parameter strategies will be required in order to go beyond these primitive attempts of exploiting the SVD in such a way as to exhibit positive phenomena in genetic algorithms.

Acknowledgments

Many thanks are due to Dr. E. Rodney Canfield for all of his wonderful guidance, insight, and support.

Bibliography

- [1] Duda, R., Hart, P.: Pattern Classification and Scene Analysis. Wiley (1973)
- [2] Flickner, M., Sawhney, H., Niblack, W., Ashley, J., Huang, Q., Dom, B., Gorkani, M., Hafner, J., Lee, D., Petkovic, D., Steele, D., Yanker, P.: Query by image and video content: The QBIC system. *Computer* **28** (1995) 23–32
- [3] Nobel, B.: Applied Linear Algebra. Prentice-Hall (1998)
- [4] Berry, M.W., Dumais, S.T., O'Brien, G.W.: Using linear algebra for intelligent information retrieval. *SIAM Review* **37** (1995) 573–595
- [5] Dumais, S.T., Furnas, G.W., Landauer, T.K., Deerwester, S., Harshman, R.: Using latent semantic analysis to improve access to textual information. In: Proceedings of ACM CHI'88 Conference on Human Factors in Computing Systems. Innovative Information Access (1988) 281–285
- [6] G.Golub, Reinsch, C.: Handbook for Matrix Computation II, Linear Algebra. Springer-Verlag (1971)
- [7] Papadimitriou, C.H., Tamaki, H., Raghavan, P., Vempala, S.: Latent semantic indexing: A probabilistic analysis. In ACM, ed.: PODS '98. Proceedings of the ACM SIGACT–SIGMOD–SIGART Symposium on Principles of Database Systems, June 1–3, 1998, Seattle, Washington, New York, NY 10036, USA, ACM Press (1998) 159–168
- [8] Ding, C.: A similarity-based probability model for latent semantic indexing. In: Proceedings of 22nd ACM SIGIRConference. (1999)
- [9] Goldberg, D.E.: Genetic Algorithms in Search, Optimization, and Machine Learning. Addison-Wesley, Reading, Mass. (1989)
- [10] Michalewicz, Z.: Genetic Algorithms + Data Structures = Evolution Programs, 3rd Revised and Extended Edition. Springer-Verlag (1999)
- [11] Khaled Rasheed, S.V., Ni, X.: Comparison of methods for using reduced models to speed up design optimization. In: The Genetic and Evolutionary Computation Conference (GECCO'2002). (2002)
- [12] Rasheed, K., Hirsh, H.: Learning to be selective in genetic-algorithm-based design optimization. *Artificial Intelligence in Engineering, Design, Analysis and Manufacturing* **13** (1999) 157–169
- [13] Berry, M.W., Drmac, Z., Jessup, E.R.: Matrices, vector spaces, and information retrieval. *SIAM Review* **41** (1999) 335–362