Dynamic and Scalable Evolutionary Data Mining: An Approach based on a Self-Adaptive Multiple Expression Mechanism

Olfa Nasraoui, Carlos Rojas, and Cesar Cardona

Department of Electrical and Computer Engineering, The University of Memphis
Memphis, TN 38152
{onasraou, crojas, ccardona}@memphis.edu

Abstract. Data mining has recently attracted attention as a set of efficient techniques that can discover patterns from huge data. More recent advancements in collecting massive evolving data streams created a crucial need for dynamic data mining. In this paper, we present a genetic algorithm based on a new representation mechanism, that allows several phenotypes to be simultaneously expressed to different degrees in the same chromosome. This *gradual multiple expression* mechanism can offer a simple model for a *multiploid* representation with self-adaptive dominance, including *co-dominance* and *incomplete dominance*. Based on this model, we also propose a data mining approach that considers the data as a reflection of a dynamic environment, and investigate a new evolutionary approach based on continuously mining non-stationary data sources that do not fit in main memory. Preliminary experiments are performed on real Web clickstream data

1 Introduction and Motivation

1.1 The need for "Adaptive Representation and Dynamic Learning" in Data Mining

Data mining has recently attracted attention as a set of efficient techniques that can discover patterns from huge data sets, and thus alleviate the information overload problem. The further advancement in data collection and measurements led to an even more drastic proliferation of data, such as sensor data streams, web clickstreams, network security data, news and intelligence feeds in form of speech, video and text, which in addition to scalability challenges, further stressed the fact that the environment in which we live is constantly changing. Thus, there is a crucial need for dynamic data mining, Specifically, within the context of data mining, there are two scenarios that call on dynamic learning:

(i) Scenario 1: The data supporting the learning task (including its nature, structure, and distribution), the goals of the learning task, or the constraints governing the feasible solutions for this task may be changing. A typical example today lies in mining sensor and data streams.

(ii) Scenario 2: The mechanism that is used to process the data for data mining may mimic the previous dynamic learning scenario. For instance, the size of the data may be huge, and thus it cannot fit in main memory, and we opt to process it incrementally, one sample at a time, or in chunks of data. In this case, there is no warranty that the different increments of data will reflect the same distribution. Hence this can be mapped to the previous dynamic learning scenario.

The type of flexibility and adaptation that is called for when learning in dynamic environments is nowhere to be found more than in nature itself. For instance, the way that DNA gets transcribed and synthesized into elaborate protein structures is dynamic. Genes get promoted and suppressed with varying degrees and in a dynamic way that adapts to the environment even within a single lifetime.

1.2 Contributions and Organization of this Paper

In this paper, we present the *Soft Structured Genetic Algorithm* (s^2GA) algorithm, and illustrate its use for non-stationary objective function optimization. We also adapt this approach to *evolutionary data mining* in non-stationary environments. s^2GA uses a *gradual multiple expression* mechanism that offers a simple model for a *multiploid* representation with self-adaptive dominance, including *co-dominance*, where both haploid phenotypes are expressed at the same time, as well as *incomplete dominance*, where a phenoptypical trait is expressed only to a certain degree (such as in certain flowers' colors).

Justifying the Choice of Multiploidy as the Underlying Adaptation Mechanism Some work on dynamic optimization has solely relied on hypermutation to recover from environmental changes [1]. Furthermore, Lewis et al. [2] have empirically shown that high mutation rates, applied when an enviroment change is detected, can outperform a simple diploid representation scheme. However, in many data mining problems, the dimensionality is extremely high, ranging in the millions in the case of web usage and gene sequence data. For example, each URL on a website can be mapped to a different attribute. This will lead to an excessive devotion of the computing resources just for the bit mutations, and slow the search process. Moreover, the comparative results in [2] were based on diploidy with a simple adaptive dominance mechanism and uniform crossover that does not take into account the arbitrary permutations of the subchromosomes within the diploid chromosome. In fact, most existing multiploidy schemes perform the crossover in a blind way between two parent chromosomes without any consideration to the important information that differentiates each subchromosome from the others. When the dominance genes are evolved together with the structural information genes, this blind crossover can be shown to cause all the chromosomes and even their subchromosomes to converge to an identical copy in the long term. This in turn defeats the purpose of multiploidy which serves primarily as a memory bank and a source of diversity. For these reasons, we present a new *specialized* crossover that avoids this problem by encouraging crossover between only the most *similar* subchromosomes, hence preserving the diversity *within each* chromosome.

Problems with the Current State of the Art in Web Usage Mining and New Contributions The majority of web mining techniques (see Section 2.2) assume that the entire Web usage data can reside in main memory. This can be a disadvantage for systems with limited main memory, since the I/O operations would have to be extensive to shuffle chunks of data in and out, and thus compromise scalability. Today's web sites are a source of an exploding amount of clickstream data that can put the scalability of any data mining technique into question. Moreover, the Web access patterns on a web site are very dynamic in nature, due not only to the dynamics of Web site content and structure, but also to changes in the user's interests, and thus their navigation patterns. The access patterns can be observed to change depending on the time of day, day of week, and according to seasonal and external events. As an alternative to locking the state of the Web access patterns in a frozen state depending on when the Web log data was collected and preprocessed, we propose an approach that considers the Web usage data as a reflection of a dynamic environment, and investigate a new evolutionary approach, based on a self-adaptive multiploidy representation, that continuously learns dynamic Web access patterns from non-stationary Web usage environments. This approach can be generalized to fit the needs of mining dynamic data or huge data sets that do not fit in main memory.

Organization of this Paper The remainder of this paper is organized as follows. We start with a background overview in Section 2. Then, in Section 3, we present a modification to the GA, based on a soft multiple Expression mechanism, for non-stationary function optimization. Based on the soft multiple Expression GA model, we present in Section 4, an evolutionary approach, called *DynaWeb*, for mining dynamic Web profiles automatically from changing clickstream environments. In Section 5, we present simulation results for synthetic non-stationary fitness functions. Then, in Section 6, we present experimental results that illustrate the performance of *DynaWeb* in mining profiles from dynamic environments on a real website. Finally, we present our conclusions in Section 7.

2 Background

2.1 Genetic Optimization in Dynamic Environments

Dynamic objective functions can make the evolutionary search extremely difficult. Some work has focused on altering the evolutionary process, including the selection strategy, genetic operators, replacement strategy, or fitness modification [3, 2, 1], while other work focused on the concept of genotype to phenotype mapping or gene expression. This line of work includes models based on diploidy and dominance [4], messy GAs [5], Gene Expression Messy GA [6], overlapping genes such as in DNA coding methods [7–9], the floating point representation [10], and the structured GA [11]. In particular, the structured GA (sGA) uses a structured hierarchical chromosome representation, where lower level genes are collectively switched on or off by specific higher level genes. Genes that are switched on are expressed into the final phenotype, while genes that are switched off do not contribute to coding the phenotype. A modification of the sGA based on the concept of soft activation mechanism was recently proposed with some preliminary results in [12]. This approach is detailed in Section 3.

2.2 Mining the Web for User Profiles

The World Wide Web is a hypertext body of close to 10 Billion pages (not including dynamic pages, crucial for interaction with Web Databases and Web services) that continues to grow at a roughly exponential rate in terms of not only content (total number of Web pages), but also reach (accessibility) and usage (user activity). Data on the Web exceeds 30 Terabytes on roughly three million servers. Almost 1 million pages get added daily, and typically, several hundred Gigabytes are changed every month. Hence, the Web constitutes one of the largest dynamic data repositories. In addition to its ever-expanding size and lack of structure, the World Wide Web has not been responsive to user preferences and interests. *Personalization* deals with tailoring a user's interaction with the Web information space based on information about him/her, in the same way that a reference librarian uses background knowledge about a person or *context* in order to help them better. The concept of *contexts* can be mapped to distinct user *profiles*. Mass profiling is based on general trends of usage patterns (thus protecting privacy) compiled from all users on a site, and can be achieved by mining user profiles from the historical *web clickstream* data stored in server access logs. A *web clickstream* is a virtual trail that a user leaves behind while surfing the Internet, such as a record of every page of a Web site that the user visits. Recently, data mining techniques have been applied to discover mass usage patterns or profiles from Web log data [13–17]. In [17], a *linear* complexity Evolutionary Computation technique, called Hierarchical Unsupervised Niche Clustering (H-UNC), was presented for mining both user profile clusters and URL associations in a *single* step. The evolutionary search allowed HUNC to exploit a subjective domain specific similarity measure, but it was limited to a stationary environment.

3 The Soft Multiple Expression Genetic Algorithm (s^{2}GA)

In the *Soft Structured Genetic Algorithm* (s^{2}GA), the lower level or structural information genes are no longer limited to total expression or to none. Instead, they can be expressed to different continuous degrees. Hence, several phenotypes can be simultaneously expressed in the same chromosome, but to different degrees. This *gradual multiple expression* mechanism can offer a simple model for a *multiploid* representation with self-adaptive dominance, including *co-dominance*, where both haploid phenotypes are expressed at the same time, as well as *incomplete dominance*, where a phenoptypical trait is expressed only to a certain degree (such as in the color of some flowers). Compared to the structured GA, in the soft activation mechanism, the activation of the subchromosomes in the lower levels is not a crisp value (active or not). Instead, every subchromosome has a soft activation/expression value in the interval $[0, 1]$. This allows the expression of multiple subchromosomes. To get this soft activation, the number of redundant subchromosomes is fixed to N_A. The dominance mechanism, traditionally used to decide the final phenotype that gets expressed is not fixed a priori, but rather adapts by evolution to express the best-fit subchromosomes depending on the current environment. The *dominance* or *activation* value for each subchromosome is controlled by a soft activation gene, A_i, a real number in the interval $[0, 1]$. The values for the soft activations are obtained as follows. In general, if there are N_A soft activation genes A_i, $i \in 1, 2, \cdots, N_A$, each encoded on l_a bits, the value a_i for the soft activation gene A_i is:

$$a_i = \begin{cases} \frac{D_i}{\sum_{j=1}^{N_A} D_j}, & \text{if } \sum_{j=1}^{N_A} D_j \neq 0 \\ \frac{1}{N_A}, & \text{if } \sum_{j=1}^{N_A} D_j = 0 \end{cases} \tag{1}$$

Where D_j is the decimal value of the l_a bits coding the A_j soft activation gene. Therefore $a_i \in [0, 1]$, and $\sum_{i=1}^{N_A} a_i = 1$. This has the advantage of keeping a chromosome with the same data encoding (binary) for both the activation and the information genes. The activation genes are constrained to sum to 1 in the preliminary model, but this constraint is not required. Hence, $\sum_{i=1}^{N_A} a_i = 1$. But they can be nonzero simultaneously. This means that *several* different expressions can co-exist in the same population, same generation, and same chromosome. It is this feature that is expected to allow for gradual adaptations of the genome to dynamic environments. The fitness computation of this genetic algorithm can consider all the subchromosome expressions in order to compute an aggregate fitness for the entire chromosome. This is accomplished by a weighted fitness. However, other aggregation mechanisms, such as the fitness of the *maximally activated* subchromosome, or the maximum of the fitnesses among the *sufficiently activated* subchromosomes, are possible. The weighted fitness is given by

$$f = \sum_{i=1}^{N_A} a_i \, f_i . \tag{2}$$

Modified Two Point Crossover In this modification, first, a usual two point crossover is made on the structural genes. The crossover points are selected such that an offspring inherits the same proportion of activation bits from the parent, as the proportion of structural bits, that is inherited. Then, a usual two point crossover is performed on the activation genes.

A New Specialized Crossover for Multiploid Chromosomes This specialization performs an independent crossover for each information subchromosome. First, a measure of the distance (the phenotypical distance) between the subchromosomes of the parents is computed, and each subchromosome from one parent is paired with the most similar unpaired subchromosome from the other parent. Next, a one point crossover between the paired subchromosomes is done (some care is taken to guarantee that all the subchromosomes participate in the crossover). Finally, the activation genes are crossed, by performing a one point crossover between each pair of corresponding activation strings (the correspondence is obtained from the matching between the paired subchromosomes).

Advantages of the soft activation mechanism The soft multiple expression and activation mechanism is expected to have the following advantages:

1. All the genotype data in the chromosome can be expressed to some degree. However, this level of expression can depend on the goodness and activation of *all* the subchromosomes.
2. The inherently redundant information, and the soft activation mechanism provide a *robust* chromosome. In order to damage the quality of the chromosome, a significant change must *concurrently* disrupt the data in the *activation and information* genes.
3. Depending on the activation values, and on how they are interpreted, more than one soft genotype can map to a single phenotype. Similarly, a single soft genotype can map to several phenotypes. This property has been lately recognized as very desirable to solve highly complex optimization problems [6].

4 DynaWeb: Mining Web Usage Data in Dynamic Environments

4.1 Extracting Web User Sessions

The access log for a given Web server consists of a record of all files accessed by users. Each log entry consists of: (i) User's IP address, (ii) Access time, (iii) URL of the page accessed, $\cdots$, etc. A user session consists of accesses originating from the same IP address within a predefined time period. Each URL in the site is assigned a unique number $j \in \{1, \ldots, N_U\}$, where N_U

is the total number of valid URLs. Thus, the i^{th} user session is encoded as an N_U-dimensional binary attribute vector $\mathbf{s}^{(i)}$ with the property

$$s_j^{(i)} = \begin{cases} 1 \text{ if the user accessed the } j^{th} \text{ URL during the } i^{th} \text{ session} \\ 0 \text{ otherwise} \end{cases}$$

4.2 Assessing Web User Session Similarity

Due to the asymmetric binary nature of the URL attributes, in this paper, we use the cosine similarity measure between two user-sessions, $\mathbf{s}^{(k)}$ and $\mathbf{s}^{(l)}$, given by $S_{kl} = \frac{\sum_{i=1}^{Nu} s_i^{(k)} s_i^{(l)}}{\sqrt{\sum_{i=1}^{Nu} s_i^{(k)}}\sqrt{\sum_{i=1}^{Nu} s_i^{(l)}}}$. Finally, this similarity is mapped to the dissimilarity measure $d_s^2(k,l) = (1 - S_{kl})^2$.

4.3 Mining Web User Profiles by Clustering Web Sessions

The proposed dynamic evolutionary Web mining algorithm, *DynaWeb* uses the s ^{2}GA algorithm in representing and evolving the population. It uses the folowing representation: Each chromosome consists of N_A subchromosomes. Each subchromosome encodes a possible session prototype or profile that consists of a binary string of length N_U URLs, with same format as the binary session attribute vectors $\mathbf{s}_i$ defined in Section 4.1. Hence, each chromosome may encode different profiles, where each profile can be expressed to a certain degree in $[0, 1]$. The cosine based dissimilarity measure, defined in Section 4.2, is used to compute the distance between session data and candidate profiles.

The fitness value, f_i, for the i^{th} candidate profile, $\mathbf{P}_i$, is defined as the density of a hypothetical cluster of Web sessions with $\mathbf{P}_i$ as a summarizing prototype or medoid. It is defined as $f_i = \frac{\sum_{j=1}^{N} w_{ij}}{\sigma_i^2}$, where w_{ij} is a robust weight that measures how typical a session $\mathbf{s}_j$ is in the i^{th} profile, and is given by

$$w_{ij} = \exp -\frac{d_{ij}^2}{2\sigma_i^2}. \tag{3}$$

σ_i^2 is a robust measure of scale (dispersion) for the i^{th} profile, d_{ij}^2 is a distance measure from session $\mathbf{s}_j$ to profile $\mathbf{P}_i$, and N is the number of data points. Note that the robust weights w_{ij} will be small for outliers, hence offering a means of distinguishing between good data and noise. The scale parameter that maximizes the fitness value for the i^{th} profile can be found by setting $\frac{\partial f_i}{\partial \sigma_i^2} = 0$ to obtain $\sigma_i^2 = \frac{\sum_{j=1}^{N} w_{ij} d_{ij}^2}{2\sum_{j=1}^{N} w_{ij}}$. To get unbiased scale estimates, the above scale measure should be compensated by a factor of 2, which results in

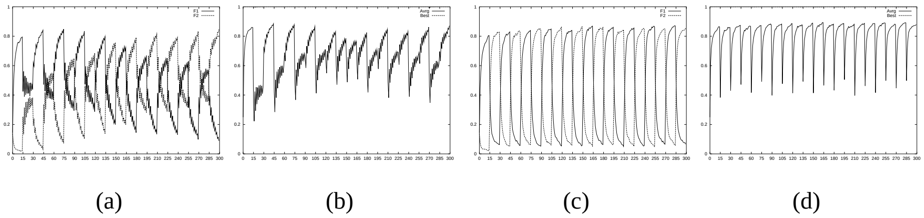
$$\sigma_i^2 = \frac{\sum_{j=1}^{N} w_{ij} d_{ij}^2}{\sum_{j=1}^{N} w_{ij}}. \tag{4}$$

Therefore, w_{ij} and σ_i^2 will be alternatively updated using (3) and (4) respectively, for 3 iterations for each individual, starting with an initial value of $\sigma_{initial}^2$, and using the previous values of σ_i^2 to compute the weights w_{ij}. This *hybrid* genetic optimization converges much faster than a purely genetic search. More details about the underlying mechanism for stationary environments can be found in [18] and [17].

5 Simulation Results for Synthetic Non-Stationary Fitness Functions

The s^{2}GA was applied to the alternating optimization of two non-overlapping objective functions, $F1$ and $F2$, defined in the interval $[0, 1]$, and each having a single peak with height $= 1$. These functions are translations of the function $F(x) = \left(\frac{27(-x^3+x^2)}{4}\right)^{10}$, and given by $F1(x) = F(0.8 - x)$ and $F2(x) = F(x - 0.2)$. The non-stationary optimization was based on periodical swappings between $F1$ and $F2$, as fitness functions, every $n = 15$ generations for a total of 300 generations. In all experiments, the population size was 200, the crossover rate was 0.9, and the mutation rates were 0.01 and 0.05 for the structural and activation bits, respectively. First, we plot the proportion of Good chromosomes (individuals that accomplish more than 80% of the optimal fitness value) for each one of the evaluated functions versus the generation number. Next, we plot the average and best chromosome performance (defined below) against the generation number. The entire procedure was repeated 30 times and average results are reported in the plots. The s^{2}GA representation consisted of 2 binary subchromosomes, each consisting of 10 structural information bits encoding a real number in $[0, 1]$. Each subchromosome was expressed by a 3-bit activation gene, resulting in a total chromosome length of 26.

The fitness function of the chromosome was defined as the weighted (by the activation values) aggregation of the fitnesses of all their subchromosomes. However, a single chromosome truly expresses different phenotypes. This led us to define the following measures: **(i) Activation threshold, α:** Sufficient activation value for considering a subchromosome as "activated". In our experimentats, we used $\alpha = 0.4$, i.e., 80% of the expected activation per gene, (i.e., $0.8(1/N_A)$ given a uniform activation distribution on N_A subchromosomes). **(ii) Subchromosome fitness:** subchromosome fitness evaluated using the current objective function. **(iii) Best Expressed Subchromosome:** subchromosome with highest subchromosome fitness among the ones with activation exceeding α. **(iv) Chromosome performance:** Fitness of the Best Expressed Subchromosome. In the new *specialized crossover*, special care is taken so that only similar subchromosomes are combined, regardless of their order inside the chromosome. From the point of view of exploitation, this recombination operator performs very well, contributing to the fast adaptation of the population to each new environment (see Figs. 1(c) and (d)).



(a) (b) (c) (d)

Fig. 1. Results for non-stationary function optimization, averaged over 30 runs (a,b) with *modified two point crossover* versus (c,d) with *specialized crossover*. (a,c) show Proportion of Good subchromosomes, while (b,d) show Average and Best Chromosome Performance.

6 Dynamic Web Usage Mining Experimental Results

The real clickstream data used in this section consists of 1703 sessions and 369 URLs extracted from Web logs of a department's website. The following experiment was performed to illustrate how an evolutionary algorithm can be used for mining dynamic data to discover Web user profiles. In order to simulate a non-stationary environment for Web mining in a controlled experiment, we used a coarse partition previously obtained and validated using H-UNC [17], and partially listed in Table 1, in order to consider the sessions that were assigned to each cluster as representing a different environment. Thus, each environment corresponds to a different Web usage trend. The sessions from these clusters were split into 20 different clickstream data sets, each one consisting of the sessions that are closest to one of the 20 profiles. The Genetic algorithm tried to evolve profiles, while facing a changing data set obtained by alternating the data from each of the 20 usage trends. The process was repeated for several epochs, each time presenting the succession of different data sets in alternation, simulating non-stationary observed usage trends.

Table 1. Summary of some usage trends previously discovered using Hierarchical Unsupervised Niche Clustering (only URLs with top 3 to 4 relevance weights shown in each profile)

i	$\lvert P_{T\,i} \rvert$	$P_{T\,i}$
0	106	{0.99 - /people_index.html}, {0.98 - /people.html}, {0.97 - /faculty.html}
1	104	{0.99 - /}, {1.00 - /cecs_computer.class}
2	177	{0.90 - /courses_index.html}, {0.88 - /courses100.html}, {0.87 - /courses.html} , {0.81 - /}
3	61	{0.80 - /}, {0.48 - /degrees.html}, {0.23 - /degrees_grad.html}
4	58	{0.97 - /degrees_undergrad.html}, {0.97 - /bsce.html}, {0.95 - /degrees_index.html}
5	50	{0.56 - /faculty/springer.html}, {0.38 - /faculty/palani.html}
6	116	{0.91 - /˜saab/cecs333/private}, {0.78 - /˜saab/cecs333}
12	74	{0.57 - /˜shi/cecs345}, {0.45 - /˜shi/cecs345/java_examples}, {0.46 - /˜shi/cecs345/Lectures/07.html}
13	38	{0.82 - /˜shi/cecs345}, {0.47 - /˜shi}, {0.34 - /˜shi/cecs345/references.html}
14	33	{0.55 - /˜shi/cecs345}, {0.55 - /˜shi/cecs345/java_examples}, {0.33 - /˜shi/cecs345/Projects/1.html}
15	51	{0.92 - /courses_index.html} , {0.90 - /courses100.html}, {0.86 - /courses.html}, {0.78 - /courses200.html}
16	77	{0.78 - /˜yshang/CECS341.html}, {0.56 - /˜yshang/W98CECS341}, {0.29 - /˜yshang}
19	120	{0.27 - /access} , {0.23 - /access/details.html}

We simulated the following *dynamic* scenarios:

scenario 1 (straight): We presented the sessions to *DynaWeb* one profile at a time for 50 generations each: sessions assigned to trend 0, then sessions assigned to trend 1, $\cdots$, until trend 19.

scenario 2 (reverse): We presented the sessions to *DynaWeb* one profile at a time for 50 generations each, but in *reverse* order: sessions assigned to trend 19, $\cdots$, until sessions assigned to trend 0.

scenario 3 (multi-trend): The sessions are presented in bursts of simultaneous multiple usage trends for 200 generation per multi-trend: First the sessions in profiles 7 and 8 are presented together for 200 generations, followed by the sessions in profiles 9 and 14, and finally by profiles 15 and 16, to test diversity as well as dynamic adaptation.

The proposed algorithm, *DynaWeb*, was applied with *specialized crossover*, a population of $N_P = 50$ individuals, initialized by selecting sessions randomly from the input data set, and with chromosome encoding based on 5 subchromosomes, each activated by one of $N_A = 5$ continuous valued activation genes. Each activation gene is encoded on 3 bits. The crossover probability was 0.9 per subchromosome, and the mutation probability was 0.01 per bit for the structural genes, and 0.05 per bit for the activation genes. The fitness of a chromosome was computed as the fitness of the subchromosome with maximum activation value in the case of scenarios 1 and 2, and as the combined fitness for scenario 3 to encourage diversity in this multimodal scenario. The ability of the population to evolve in a dynamic way when facing each new environment was evaluated in each generation by comparing the *good* individuals in the population to the ground-truth profiles, P_{Ti}, ($i = 0, \cdots, 19$). To do this, we defined as *good* individuals, those individuals that have a combined fitness exceeding $(f_{max} + f_{avg})/2$, where f_{max} and f_{avg} are the maximal and average fitness in the current generation, respectively. Before comparing an individual to the ground truth profiles, an expressed phenotype must first be extracted. In our case, the active (i.e., with activation gene value $> \alpha$) subchromosome with highest fitness, was used to yield the final expressed phenotype. It is this phenotype that is compared with each of the ground-truth profiles in each generation. We do this by computing the cosine similarity between the phenotype expressed by each *good* chromosome and each of the ground-truth profiles, P_{Ti}, $i = 0, \cdots, 19$. The similarities computed using all the good chromosomes are averaged in each generation, to yield measures $\hat{S}_i$ for each ground-truth profile, P_{Ti}, $i = 0, \cdots, 20$. These measures are used to assess whether the evolution is able to adapt to each change in the environment. Ideally, adaptation to the i^{th} environment is quantified by the fact that $\hat{S}_i$ gradually becomes higher than all other $\hat{S}_j, j \neq i$.

The above procedure was repeated 20 times and the results are averaged. Stochastic *Viral injection/replacement* was used. This phenomenon is different from traditional evolutionary techniques, in that genetic material from an external organism gets injected into the host organism's DNA. It is common with viruses such as the AIDS virus. Given the nature of our *data driven* approach, it is expected that this operation will refresh the current genome with vital and current information from the new environment. This step stochastically replaced with a 0.3 injection rate per generation the most active subchromosome from the worst individual of the current population (based on their combined chromosome fitness) with data randomly selected from the data set being presented in the current generation. The results for *scenario 1: straight order* are shown in Fig. 2 and Fig. 3, for *DynaWeb* and the *Simple GA*, respectively. Fig. 2, *which is better viewed in color*, shows that as each environment comes into context, the genomes in the current population gradually evolve to yield candidate profiles that match the new environment. That is, whenever the environment changes from j to i, the similarity measure that is the highest gradually switches from being $\hat{S}_j$ to becoming $\hat{S}_i$. Hence, the genome succeeds in tracking the dynamic web usage trends, which is the desired goal. We have also observed a successful adaptation of the expression/activation genes, switching between different parts of the chromosome to track the changing environments. We note that the average similarity, $\hat{S}_i$, achieved for certain usage environments (such as profile 19) are relatively low. This is because the sessions in these environments have more variability, contain more noise, and thus form a less compact cluster, as can be judged by their lower URL relevance weights in Table 1. Fig. 2 also shows a desired property in the cross-reaction between overlapping usage trends. For example the first 5 usage trends overlap significantly since they represent outside visitors to the website, mostly prospective students, with slightly different interests. Fig. 3 shows that the simple GA yields a population that is too slow to adapt, and with lower quality.

The results using *DynaWeb* for *scenario 2: reverse order* and for *scenario 3: multi-trend* are shown in Figure 4 and Figure 5, respectively. Figure 4 shows that the order of presentation of the environments is not important, since it is merely a vertical reflection of the evolution for scenario 1. Figure 5 shows the ability of *DynaWeb* to track multiple profiles simultaneously, even as they change. Except for the first epoch, the remaining epochs show a consistent adaptation to the presented usage trends, since the population achieves highest similarity to the two current usage trends, as compared to the remaining 4 trends. The improvement in adaptation starting from the second cycle shows the presence of a good *memory* mechanism that is distributed over the different subchromosomes of the population, a memory that comes into context, i.e. becomes expressed when it is relevant in the current context, and goes dormant in other contexts.

7 Conclusion

For many data mining tasks, the subjective objective functions and/or dissimilarity measure may be non-differentiable. Evolutionary techniques can handle a vast array of subjective, even non-metric dissimilarities. We proposed a new framework that considers evolving data, such as in the context of mining stream data, as a reflection of a dynamic environment which therefore requires dynamic learning. This approach can be generalized to mining huge data sets that do not fit in main memory. Massive data sets can be mined in parts that can fit in the memory buffer, while the evolutionary search adapts to the changing trends automatically. While it is interesting to compare the proposed approach against other standard dynamic optimization strategies, one must keep in mind that *domain knowledge*, *scalability*, and a *data-driven* learning framework are crucial to most real life data mining problems, and this in turn may require *nontrivial* modifications to most existing techniques including those that are based on adaptive case-based memories, hypermutation, and simple dominance schemes.

Acknowledgment

This work is supported by the National Science Foundation (CAREER Award IIS-0133948 to O. Nasraoui).

References

1. H. G. Cobb, “An investigation into the use of hypermutation as an adaptive operator in genetic algorithms having continuous, time-dependent nonstationary environments,” Tech. Rep. AIC-90-001, Naval Research Laboratory, Washington, 1990.
2. J. Lewis, E. Hart, and R. A.Graeme, “A comparison of dominance mechanisms and simple mutation on non-stationary problems,” in *5th International Conference on Parallel Problem Solving from Nature*, 1998, pp. 139–148.
3. J. Branke, “Evolutionary approaches to dynamic optimization problems: A survey,” *Evolutionary Algorithms for Dynamic Optimization*, pp. 134–137, 1999.
4. D. Goldberg and R. E. Smith, “Nonstationary function optimization using genetic algorithms with diloidy and dominance,” in *2nd International Conference on Genetic Algorithms*, J. J. Grefensette, Ed., Lawrence, 1987, pp. 59–68.

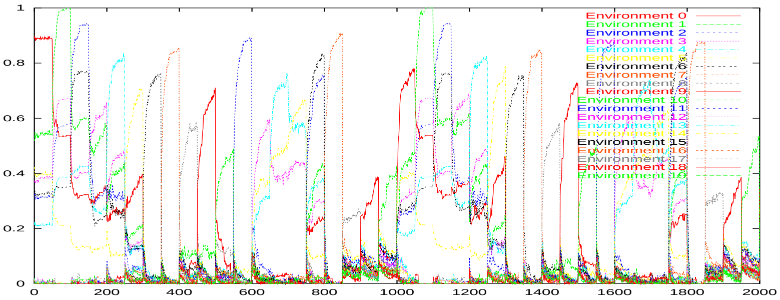


Fig. 2. Average similarity to ground-truth profiles among good individuals averaged for 20 runs, for scenario 1 with DynaWeb, $N_A = 5$ subchromosomes, 0.3 injection, for scenario 1 (Straight order of usage trends)

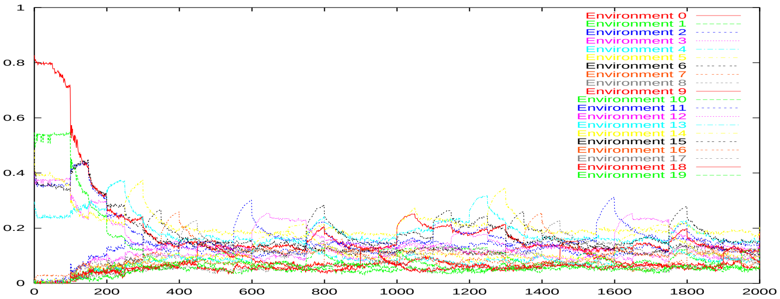


Fig. 3. Average similarity to ground-truth profiles among good individuals averaged for 20 runs, for scenario 1 with the Simple GA

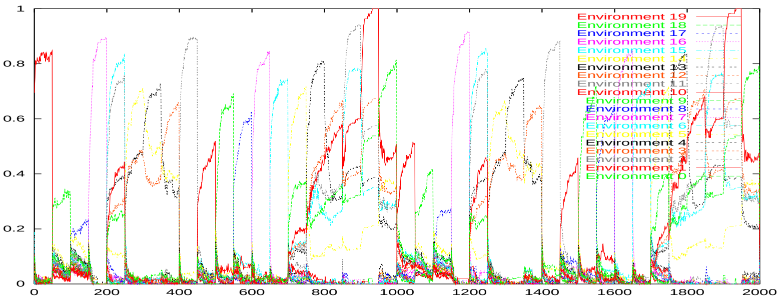


Fig. 4. Average similarity to ground-truth profiles among good individuals averaged for 20 runs, for DynaWeb with $N_A = 5$ subchromosomes, 0.3 injection, for scenario 2 (Reverse order of usage trends)

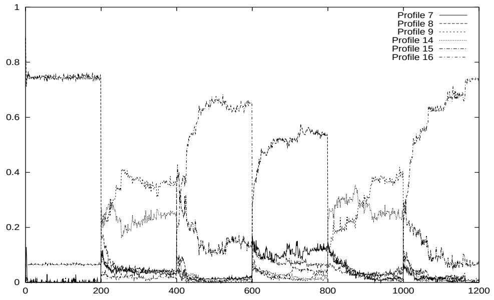


Fig. 5. Average similarity to ground-truth profiles among good individuals averaged for 30 runs, for DynaWeb with $N_A = 5$ subchromosomes, 0.3 injection, for scenario 3 (alternating multi-usage trends)

5. D. Goldberg, K. Deb, and B. Korb, “Nonstationary messy genetic algorithms: motivation, analysis, and first results,” *Complex Systems*, vol. 3, pp. 493–530, 1987.
6. H. Kargupta, “The gene expression messy genetic algorithm,” in *International Conference on Evolutionary Computation*, 1996.
7. W. Wienholt, “A refined genetic algorithm for parameter optimization problems,” in *5th International Conference on Genetic Algorithms*, 1993.
8. D. K.Burke, J. DeJong, C. Grefensette, and A. Wu, “Putting more genetics into genetic algorithms,” *Evolutionary Computation*, vol. 6, no. 4, 1998.
9. A. Wu and R. K. Lindsay, “Empirical studies of the genetic algorithm with non-coding segments,” *Evolutionary Computation*, vol. 3, no. 2, pp. 121–147, 1995.
10. A. Wu and R. K. Lindsay, “A comparison of the fixed and floating building block representation in the genetic algorithm,” *Evolutionary Computation*, vol. 4, no. 2, pp. 169–193, 1996.
11. D. Dasgupta and D. McGregor, “Nonstationary function optimization using structured genetic algorithm,” in *Parallel Problem Solving For Nature Conference*, Belgium, 1992.
12. O. Nasraoui, C. Rojas, C. Cardona, and D. Dasgupta, “Soft adaptive multiple expression mechanism for structured and multiploid chromosome representations,” in *Genetic and Evolutionary Computation Conference, late breaking papers*, Chicago, July 2003.
13. O. Zaiane, M. Xin, and J. Han, “Discovering web access patterns and trends by applying olap and data mining technology on web logs,” in *Advances in Digital Libraries*, Santa Barbara, CA, 1998, pp. 19–29.
14. M. Perkowitz and O. Etzioni, “Adaptive web sites: Automatically synthesizing web pages,” in *AAAI 98*, 1998.
15. R. Cooley, B. Mobasher, and J. Srivastava, “Data preparation for mining world wide web browsing patterns,” *Knowledge and Information Systems*, vol. 1, no. 1, 1999.
16. O. Nasraoui, H. Frigui, R. Krishnapuram, and A. Joshi, “Mining web access logs using relational competitive fuzzy clustering,” in *8th International World Wide Web Conference*, Toronto, Canada, 1999.
17. O. Nasraoui and R. Krishnapuram, “A new evolutionary approach to web usage and context sensitive associations mining,” *International Journal on Computational Intelligence and Applications - Special Issue on Internet Intelligent Systems*, vol. 2, no. 3, pp. 339–348, 2002.
18. O. Nasraoui and R. Krishnapuram, “A novel approach to unsupervised robust clustering using genetic niching,” in *Ninth IEEE International Conference on Fuzzy Systems*, San Antonio, TX, May 2000, pp. 170–175.