

# Design and Implementation of a Genetic-Based Algorithm for DataMining

Sunil Choenni

Presented by:  
Sam Beskur  
Aaron Eagerman

## Overview

- Problem
- Introduction to genetic algorithms
- Key contributions
- Prior work
- Methodology
- Results
- Summary & Opinion

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Problem

- Data mining deals with very large search spaces. It is not feasible to perform an exhaustive search.
- Success of a strategy depends on the search space. This information is not always known in data mining.
  - Few optima? Many Optima?

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Introduction to Genetic Algorithms

- Individuals
- Population
  - Set of individuals
- Fitness function
- Manipulation
  - Mutation
  - Cross-over

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Key Contributions

- Implement a genetic algorithm
  - SHARVIND is a prototype re-targetable data mining tool that is equipped with a genetic algorithm
- Optimization rules to improve the efficiency of the search
- Represent individuals as expressions instead of binaries
- Intelligent mutation operations
- Mined 2 real databases with SHARVIND

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Prior Work

- Proposed genetic algorithms for data mining
- Focus on machine learning
- Hill climber

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Methodology - Individuals

- Where clause gender is '1' ^ age in [10,20]
- Expression instead of binary representation
  - Map directly to the database
- Must be useful
  - Does not select 0 tuples height is '10' ^ height is '12'
  - Select tuples with common characteristics
  - Contain at least one conjunction otherwise the results are likely to be trivial

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Methodology – Fitness Function

- Primary focus of a genetic algorithm
- Should represent issues that are important to a specific problem
  - Cover how well an individual represents a target
    - Cover of target class ( $\alpha$ )
      - Not interested if a target class is not represented by a significant portion of the database
    - Ratio of individual cover to cover of target class ( $\beta$ )
      - ~0 very few tuples in the target class are represented by the individual
      - ~1 perfect cover, but likely to be trivial information

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Methodology – Manipulation by Mutation

- Changes an individual into a modified version of itself
- Mutate differently based on attribute types
  - No ordering relationship town in ('a', 'b', 'c')
    - Randomly select a value and replace with a different one
  - Ordering relationship with single value
    - Change the value by  $\delta$
  - Ordering relationship with range of values
    - Randomly select the low or the high and change it by  $\delta$ 
      - [10,20] --> [10,34]

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Methodology – Manipulation by Crossover

- Takes 2 individuals and forms 2 new individuals
- Select a random point in the individual and swap the expressions before and after that point
 

Original: (a1^a2^a3^a4^a5) (b1^b2^b3^b4^b5)

New: (a1^a2^a3^b4^b5) (b1^b2^b3^a4^a5)

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Methodology – Algorithm Optimizations and Approaches

- Similar individuals
  - Can determine if a new individual's fitness is guaranteed to be  $\leq$  to another.
  - If the fitness is guaranteed to be less, then mutate it before computing its fitness. Fitness computations can be expensive.
- Elitist recombination
  - Parent and offspring are compared against each other. The most fit parent and child are chosen for the next population
    - No need for crossover probability
    - No intermediate population

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001

## Methodology - Algorithm

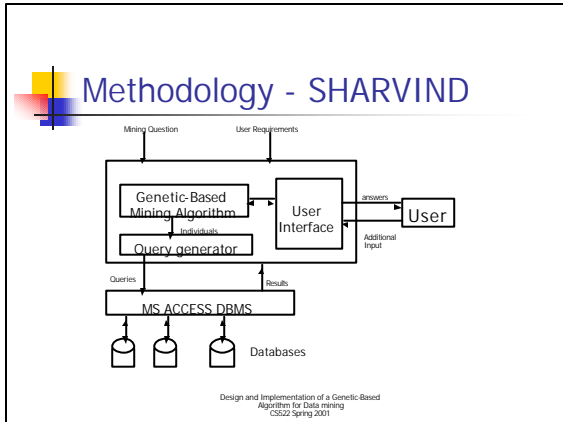
```

Initialize a population P(t)
Find the fitness values of the individuals in P(t)
While (F(P(t+1)) - F(P(t)) >= e)
  shuffle the population
  foreach group of 2 consecutive individuals
    cross-over the parents to form 2 children
    select the parent with the higher fitness and discard the other
    mutate the selected parent with probability c
    put the selected parent into the new population P(t+1)
  foreach of the children
    if it is a similar of the selected parent and its fitness is lower mutate the child with probability 1, otherwise mutate with probability c
    put the child with the highest fitness into P(t+1)

compute the total fitness of P(t) and P(t+1) as the sum of the individual fitness values
  
```

P(t) = population at time t  
 P(t+1) = next population  
 e = value by which the next population's fitness must improve on the previous

Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CSS22 Spring 2001



- ## Results – Test Database
- Known Results
  - Varied initial populations
    - Random
    - Modified random – known good values replaced by known bad values
    - Bad – individuals with low fitness values
  - Found near optimal solutions for all three cases
- Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CS522 Spring 2001

- ## Results – Aircraft Incident Databases
- ECCAIRS and FAA are real databases that contain flight incident data. The authors tried to mine characteristics of unsafe flights.
  - Extensive data cleaning. Trimmed attributes from 186 to 64 and 70 to 30 in the respective databases.
  - Used answers to original mining question to derive further questions.
  - Findings were presented to a flight expert
    - Mostly known and/or trivial information but helped to gain insight into the data.
    - The expert did not realize that student information was stored in the database.
    - Expert said that results were accurate and promising
- Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CS522 Spring 2001

- ## Summary & Opinion
- No comparison to other algorithms.
  - There is no information about how other algorithms would perform on the test data, either before or after cleaning.
  - Value comes from showing an actual implementation.
  - They proved that a genetic algorithm can be used to state information about databases.
  - Bottom Line: this paper does not add very much information to the field.
- Design and Implementation of a Genetic-Based Algorithm for Data Mining  
CS522 Spring 2001

# Design and Implementation of a Genetic-Based Algorithm for DataMining

---

Sunil Choenni

Presented by:  
Sam Beskur  
Aaron Eagerman