Data Mining Using:
Genetic Algorithms, Tabu Search, Rule Induction, Decision Trees, Clustering, and Nearest Neighbor

Dr. Hamid R. Nemati
ISM 611

Presentation Outline

- Knowledge Discovery Through Data Mining
- Genetic Algorithms for data mining defined
- Tabu Search for data mining defined
- Rule Induction for data mining defined
- Rule Extraction from Neural Networks
- Decision Trees for data mining defined
- Clustering and Nearest Neighbor for data mining defined

Genetic Algorithms

- Genetic algorithms refer to methods that simulate evolutionary systems.
- GA are algorithms that dictate how populations of organisms should be formed, evaluated, and modified.
- They are techniques referred to as a metaheuristic.
- Genetic algorithms avoid complete enumeration of the solution space by using ‘rules of thumb’ to find a good solution.

How can GA be used?

- There are three main areas to apply GA:
  - Optimization.
    - Optimization problems attempt to find the best solution(s) for a given problem that has several parameters (goals or resources) with associated constraints. GA can be used to automatically determine the optimal values for the variables that optimize the profit.
    - GA does not guarantee the very best solution.
  - Prediction.
    - GA is used as metalevel operators that are used to help optimize other data mining algorithms. For instance, they have been used to optimize the weights in a neural network or to find the optimal association rules in market-basket analysis.
  - Simulation.
    - When a specific business problem is not well defined, GA can be used to simulate via simple interaction rules over time.

Where to Use Genetic Algorithms

- Four key attributes for a business problem that could benefit from the application of genetic algorithms:
  1. The value from various proposed solutions can be well defined.
  2. The problem is complex and cannot be solved directly.
  3. The problem is relatively new and not well understood, and no one has yet been able to determine other optimization techniques to be used for its solution.
  4. The problem involves a large number of variables working together but can be modeled through simpler processes working together to produce a large-scale effect.

Benefits of GA

- The solution time is very predictable, and is not radically affected as the problem gets larger.
- Handles non-linear and discontinuous functions equally as well as linear and continuous.
- You need only to be able to describe a good solution, you do not need to know how to build it. Thus, it does not require heavy use of expert knowledge.
- Can produce novel results among a set of good solutions. “I would have never thought of that one!”
- Tend to be compact, containing only the fitness function and a little code to handle the GA functions.
- Can usually be embedded easily, and are easy to hybridize.
Disadvantages of GA

- Mapping of problem into GA methodology is very hard
- It is a heuristic - it does not guarantee the optimal solution.
- Since GA’s only drive toward the optimal solution using the fitness function, there is no explanation about how one might logically arrive at the solution.

Genetic Algorithms

- Essentially, providing the GA with a full solution allows it to 'know' something about all of the dimensions of the problem. The GA takes these into account when selecting which solutions to exchange elements.
- The real power of a GA is the way it uses the variables and constraints to find a solution that optimizes the objective.
- Rather than spell out each element in isolation, I will provide a flow of how the GA works, and describe the activities within their proper sequence.

Genetic Algorithms

- They can be combined with hill-climbing techniques to speed up the search process, but are able to 'jump' from local maxima because the elements that promote a good solution are being mixed up in subsequent iterations.
- The idea of 'mutation,' discussed in the next section also makes sure that elements that were left out of the initial population have a chance of being incorporated later.

Holland invented the notion of a GA 'schema' to explain why the GA is so effective.

- Simply stated, schemata are similarity subsets (sets of strings that have one or more features in common), and building blocks are those schemata that are
  1) consistently emphasized by selection and
  2) respected and exchanged by the genetic operators.

Elements of Genetic Algorithm

- Variables:
  - Resources or other parameters that are elements of the decision.
- Constraints
  - Limitations associated with the variables.
  - From a mathematical perspective, constraints slice up the solution space to further restrict the range of objective values.
- Objective
  - The goal of the problem, or an evaluation of how well the problem has been solved.

The Overall Process

1. Define the problem to be solved, providing a way to encode the problem in a chromosome and a way to measure the goodness of the solution encoded in the chromosome.
2. Initialize a population of chromosomes with random values.
3. Evaluate the fitness of each organism in the population using the previously defined fitness function.
4. Allow the multiple copies of the genetic material of the best chromosomes to be made, and delete the organisms that are less fit.
The Overall Process

5. Allow the new population of organisms to undergo mutation and sexual reproduction.
6. Evaluate the fitness of each organism in the population using the previously defined fitness function.
7. Stop if any of the following criteria are met:
   a. A solution has been found that is good enough.
   b. The system has run for the pre-specified number of generations.
   c. The system has stopped making progress towards improvement.
8. If no stopping criterion is met, then return to step 4.

A Step by Step Explanation of the GA Heuristic

Create Initial Population
- The GA selects an initial chromosome population of a specified size randomly. In other words, it fills in the input variables randomly with acceptable values. A full set of input variables is a chromosome, and it makes as many of these as you specify. For example, if the user has requested 20 population members in the problem definition, they are created by assigning random values to genes, based on the range set when the genes were defined. This provides an initial group of 20 population members for generation 0.

Decode the Chromosome
- The GA then evaluates the 'fitness' of each chromosome by finding out how well it meets the fitness function. That is, how optimal an answer does this set of input values produce?

Order the Chromosomes
- At the beginning of each generation, the population members are evaluated and then ordered according to their fitness.

Choose Which Chromosomes will Mate
- In order for crossover to occur, we must pair two population members so that genes can be exchanged.
- Mate selection is carried out using evolutionary principles.
- That is, members with the best fitness are given a higher likelihood of mating, which increases the chances of superior offspring.
- Mate selection is accomplished by using a "graduated" roulette wheel. Each member is allowed to mate once. Its mate is selected by "spinning" the roulette wheel, with the added stipulation that a member of the population is not allowed to mate with itself.

Perform Crossover
- Once a member of the population and its mate have been chosen, it must be determined which genes will be exchanged.
- This is done by randomly selecting two "cut points" in the string of genes. Genes in between the two cut points will be swapped between the population member and its selected mate. (Note: Permutation Crossover is different, and is designed to guarantee that each offspring is a valid permutation).

Store Offspring
- Each member of the original population will be given a chance to mate according to the "roulette" selection outlined above. This will result in a new population which is the same size as the original population.

Mutate Selected Chromosomes
- After the new population has been created, randomly selected members of the population will undergo mutation based on the settings made by the user.
- For a random mutation, the GA randomly selects the population members to undergo mutation according to a specified mutation probability.
- As in nature, most mutations will probably produce scary results. For this reason, the mutation probability should be kept relatively low.
- The GA then randomly selects the genes which will be mutated according to the specified probability.
- Next, each gene selected will be mutated using a random number and a specified range.

Replace parts of population with superior mutations and superior prior generation members
- The fitness for each member will be recalculated after all the genes have been mutated.
- The population members are then ordered according to their fitness.
- The best members of the old population may be added to the new population in some algorithms, unless the best members of the old population are not as good as the worst members of the new population.
- A genetic algorithm that keeps one or more of the best members from each generation is said to incorporate "elitism".
A Step by Step Explanation of the GA Heuristic

- This keeps the best members of the population from getting worse from one generation to the next, and insures that the fitness of the best member can only improve or stay the same.
- It also gives the “elite”, highest-fitness individuals further opportunities to produce offspring in subsequent generations.

- Create new generation
  - Once this has been done, the population is ready to create another generation.
  - The population will cycle through generations until the Exit Condition is met.

GA Parameter Definitions

Fitness or Evaluation Function
- Like any other optimization technique, you must specify whether you want to Minimize the function value, Maximize it, or come as close as possible to matching a value you specify.
- Population Size
  - The number of chromosomes you will work with per generation.

GA Parameter Definitions

- Gene groups
  - The variable parameters in your optimization function. A chromosome, or member of the population, is made up of an array containing values for each of the variable parameters.

- Eligible Chromosome Reproduction Number
  - The number of chromosomes you will keep and allow to reproduce after each generation.

Crossover Operator

- Crossover refers to the process of creating a new offspring trial solution by combining gene values from two parents selected from the current generation.
- There are typically at least three basic choices for this option: No Crossover, Two-point Crossover and Permutation Crossover.
- No Crossover can be selected regardless of the type of genes and the population will be improved by mutation alone.
- Two-point crossover is used with Integers and Real Number genes. Permutations have their own form of crossover.

Mutation in GA

- No Mutation
  - If No Mutation is selected, population members will only be changed from one generation to the next by crossover processes. When No Mutation is selected, it is necessary to have a large enough population to ensure that all possible gene values are present at the outset.

- Random Mutation
  - If random mutation is selected, a number of population members will have the values of some of their genes changed every generation.

Mutation

- Random bit mutation.
  - The genetic material is viewed as bits from a computer science perspective and for each bit in each organism a coin is tossed. If it comes up heads, then the bit is flipped. For instance, the allele for a given gene could be the number 7. In binary this would be 111; if the second bit were mutated, the new value would be 101, which is the number 5.

- Random gene mutation.
  - In this case the allowable alleles for a given gene are known and the entire gene is mutated to some other valid value. For instance, the hair color gene might have five values (brown, blond, red, black, auburn). In the chromosome these might be encoded as the numbers (0,1,2,3,4), which would require 3 bits to represent (000, 001, 010, 011, 100). These non-numerical alleles could be weeded out through selection, but it would be more efficient to have a more sophisticated gene-level mutation.
**Tabu Search Methodology**
- Tabu Search is an **intelligent** search meta-heuristic.
- Has close ties to genetic algorithms.
- Based on two AI principals:
  - **Adaptive Memory**
  - **Responsive Exploration**
- Once an area of the solution space is searched, it becomes TABU or forbidden for a pre-specified duration.

**Tabu Search Algorithm**
- Identify all possible moves (neighborhood) and select the best move.
- Check a **Tabu status** for the move, if the move is non-tabu, execute the move.
- If the move is tabu and satisfies an **aspiration level**, make the move.
- Repeat the process of generating neighborhood and evaluating moves until either the optimum solution is reached or a stopping criteria is satisfied.

**Tabu Search Algorithm**
- Consider a function $f(x)$ to be optimized over a set $X$.
- Each $x \in X$ has an associated neighborhood $N(x) \subseteq X$.
- Each solution $X^*$ elements of $N(x)$ can be reached from $x$ by an operation called a move.
- A Tabu Status is associated with every move. The Tabu Status can be based on recency and/or frequency of a move.

**Rule Induction**
- Rule induction is one of the major forms of data mining and is perhaps the most common form of knowledge discovery in unsupervised learning systems.
- The rules that are pulled from the database are extracted and ordered.

**What is a rule?**
- A rule has two parts:
  - The left side is called the antecedent and the the consequent.
  - The antecedent can consist of just one condition or multiple conditions which must all be true in order for the consequent to be true at the given accuracy.
  - Generally the consequent is just a single condition (e.g., prediction of purchasing just one grocery store item) rather than multiple conditions. Thus rules such as “If $x$ and $y$, then $a$ and $b$ and $c$ are uncommon.”
Need for Rule Extraction Methods

- Data Exploration and Data Mining
- Problems Associated with the "Black Box" Approach.
- Reasoning and Explanation Capabilities
- Knowledge Acquisition problems
- Initial Knowledge Domain Refinement Problems
- Cross Referencing and Verification Capabilities
- Generalization of the Network Solutions

Considerations in Developing Rule Extraction Method

- Translucency of the Extracted Rules
- Comprehensibility of the Extracted Rule (the number of rules and/or number of premises in each extracted rule)
- Accuracy of the Extracted Rules
- Portability of the Extracted Rules
- Granularity of the Explanation Feature
- Representation Format

How to evaluate the rules

1. **Accuracy** - answers the question of how often the rule will supply the correct prediction
2. **Coverage** - answers the question of how often the rule can be used for prediction (and how sensitive it would be to the sample taken to create it)
3. **Support** - answers the question of how frequently the rules (both antecedent and consequent) occur
4. **Significance** - tells the user how unlikely this pattern would be compared to random chance of independent events
5. **Simplicity** - helps the user build intuitions and confirm the rule via intuitions
6. **Novelty** - helps the user find rules that may occupy regions of the predictive space where few other rules are

How to construct rules

1. Preprocess the data so that each predictor has well-defined intervals rather than continuous values.
2. Generate initial rules from the data of just one constraint.
3. From the records, generate rules that have an additional constraint from the given rules.
4. Keep the group of rules that are good candidates to have added constraints.
5. Continue adding constraints onto the rules until the stopping criteria have been met for all rules.
6. Organize the rules on the basis of their usefulness (i.e., accuracy, coverage, support, significance, simplicity, and novelty) to the end user.

The Simplest method for generating rules

1. Generate all predictor/value pairs for each record as the first set of rules.
2. Count the number of occurrences for each rule and the antecedent by itself.
3. Calculate the accuracy, coverage, and support and eliminate those rules that do not pass the required minimum threshold.
4. For each record, see which rules apply to it and add an additional constraint (predictor value) to the rule from the record.
5. Return to step 1 until the rules are too complex or no rule passes the minimum coverage or support thresholds.

Categorized Input-Output Rule Extraction Method (CRE)

- Aim here is to find the rules that can directly map the inputs into outputs
- It is architecture independent
- It does not require any specific training regime (supervised or unsupervised)
- The network inputs and outputs have to categorized
Categorized Input-Output Rule Extraction Method (CRE)

- Assign intervals to all input and output variables.
- For each interval, define a binary variable.
- Match each output pattern of the network with each input pattern.
- Generate and refine a truth table by concatenating each input pattern with its corresponding output.
- Generate a Boolean function from the final truth table.

CRE Method Illustrated

<table>
<thead>
<tr>
<th>Years Exp</th>
<th>Years Edu</th>
<th>Salary</th>
</tr>
</thead>
<tbody>
<tr>
<td>3</td>
<td>12</td>
<td>$20</td>
</tr>
<tr>
<td>3</td>
<td>14</td>
<td>$40</td>
</tr>
<tr>
<td>2</td>
<td>18</td>
<td>$50</td>
</tr>
<tr>
<td>10</td>
<td>10</td>
<td>$30</td>
</tr>
</tbody>
</table>

- Salary: $20 or less, $40 or less, $60 or less
- Years Exp.: 2 or less, 5 or less, 10 or less
- Years Edu.: 12 or less, 16 or less, 18 or less

Considerations In Selection of Rule Extraction Method

- Transparency of the Extracted Rule
- Granularity of the Explanation Feature
- Representation Format
- Fidelity of the Extracted Rule (capability of the extracted rule to mimic the embedded knowledge in a trained network)

Generate Truth Table and the rules

<table>
<thead>
<tr>
<th>Yrs Exp</th>
<th>Yrs Edu</th>
<th>Sal</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Rule 1: If Yrs Exp is 5 or more and Yrs Edu is 16 or less
Then Salary is $40,000 or less

Rule 2: If Yrs Exp is 2 or less and Yrs Edu is 18 or more
Then Salary is $60,000 or more

Considerations In Selection of Rule Extraction Method

- Comprehensibility of the Extracted Rule (the number of rules and/or number of premises in each extracted rule)
- Accuracy of the Extracted Rules
- Portability of the Extracted Rules

Neural Networks and Rule Extraction

- Given a trained Neural Network and the training set, produce a **Concise and Accurate Symbolic Description** of the implied knowledge embedded in it.
- Rules extracted can be in the form of:
  - IF [not] input-variable [and] input variable THEN consequence
Neural Network Model Development Phases

- The Representation Phase:
  - the initial domain knowledge is represented in a symbolic format
- The Mapping Phase:
  - this knowledge is mapped into an initial connectionist architecture

Rule Extraction from Neural Networks

- Alleviates the Knowledge Acquisition problems
- Alleviates the Initial Knowledge Domain Refinement Problems
- Provides Reasoning and Explanation Capabilities
- Supports Cross Referencing and Verification Capabilities
- Supports Data Exploration and Induction of New Theories
- Improves the Generalization of the Network Solutions
- Alleviates Problems Associated with the “Black Box” Approach.

Types of Rule Extraction Methods

- Link Rule Extraction Techniques
- Black-box Rule Extraction Techniques

Link Rule Extraction Techniques (LRE)

- Focus of all these techniques are on extracting rule at the individual neurons.
- All these techniques search for the weighted links that cause a node (hidden or output) to become activated

- Then these combinations of weighted links are used to generate symbolic rules
- Usually very Heuristic based method to bound the search space for rules
- E.g., KT, Subset, M of N, and NeuroRule
Black-box Rule Extraction Techniques (BRE)

- All these techniques view the rule extraction as a learning task where the target concept is the function computed by the network and the input features are simply the network’s input features.

- Aim here is to find the rules that can directly map the inputs into outputs
- Extract rules from neural networks regardless of the type or the structure of the neural network
- E.g., BRAINNE, DEDEC, RuleNET and RULEX

Decision Trees

- A decision tree is a predictive model that can be viewed as a tree.
- Each branch of the tree is a classification question, and the leaves of the tree are partitions of the data set with their classification.
- In a DM perspective, decision trees can be viewed as creating a segmentation of the original data set.
- Each segment would be one of the leaves of the tree.
- Segmentation of customers, products, and sales regions is something that marketing managers have been doing for many years.

Decision Trees for Data Exploration

- The decision tree technology can be used for exploration of the data set and business problem.
- This is often done by looking at the predictors and values that are chosen for each split of the tree.
- Often these predictors provide usable insights or propose questions that need to be answered.

Decision Trees for Data preprocessing

- Another way that the decision tree technology has been used is for preprocessing data for other prediction algorithms.
- Because the algorithm is fairly robust with respect to a variety of predictor types (e.g., number, categorical) and because it can be run relatively quickly, decision trees can be used on the first pass of a data mining run to create a subset of possibly useful predictors that can then be fed into neural networks and nearest-neighbor and normal statistical routines—which can take a considerable amount of time to run if there are large numbers of possible predictors to be used in the model.

The General Idea in Decision Trees

- Decision trees are built from historical data.
- They are a form of supervised learning, although they are often used for exploratory analysis as well.
- The first step in the process is that of growing the tree.
- Specifically, the algorithm seeks to create a tree that works as perfectly as possible on all the data that is available.
- There is always noise in the database to some degree (there are predictors that are not being collected that have an impact on the prediction target).
Creating a Decision Tree

- To build decision trees, most algorithms look at all possible distinguishing questions that could break up the data set into segments that are homogeneous with respect to the prediction values.
- How decision trees pick one particular predictor over another one to make a split on the data set.
- Decision trees need to make some kind of a choice since only one predictor is used at each branch-point of the tree.

How to decide on a split at each branch

- Create a numeric measure of goodness of split that seems to map in a reasonable way to what they are looking for in terms of decreasing the disorder of the dataset.
- Some decision tree algorithms may use heuristics in order to pick the questions or even pick them at random.

Methods for Constructing the Decision Tree

- **Classification and Regression Trees (CART)**
  - It builds a binary tree by splitting the records at each node according to a function of a single input field. The best split is defined as one that does the best in terms of separating the records into groups where a single class dominates. (Diversity Index).
- **Chi Square Automatic Interaction Detector (CHAID)**
  - It is based on the statistical method of detecting statistical relationships between variables.
  - It build contingency tables uses the Chi Sq. test to detect interactions
- **ID3 (C4.5):** Based on the concept of entropy.

What is the optimum size of the tree?

- If the decision tree algorithm does not stop at some point, it will create more and more questions and branches in the tree so that eventually there was only one record in the segment.
- To let the tree grow to this size is computationally expensive and also unnecessary.
- Most decision tree algorithms stop growing the tree when one of three criteria are met:
  1. The segment contains only one record or some algorithmically defined minimum number of records.
  2. The segment is completely organized into just one prediction value.
  3. The improvement in organization is not sufficient to warrant making the split.

Clustering

- **Clustering** is a method in which like records are grouped together.
- Usually this is done to give the end user a high-level view of what is going on in the database.
- A simple example of clustering:
  - Most people perform clustering when they do the laundry: grouping the permanent-press, dry cleaning, whites, and brightly colored clothes is important because they have similar characteristics.
  - And, it turns out, they have important attributes in common about the way they behave (and can be ruined) in the wash.

Why Clustering Hard?

- Deciding on what constitute a cluster in a dynamic environment is very difficult.
- There are many different attributes that can be used to define each cluster.
- Consider the following cars to be segmented into different clusters:
  - Audi
  - BMW
  - Honda Civic
  - Saturn
  - Toyota
  - Lexus
  - Infinti
  - Buick
Nearest Neighbor

- Nearest-neighbor techniques are among the easiest to use and understand because they work in a way similar to the way people think-by detecting closely matching examples.
- Nearest neighbor is a prediction technique that is quite similar to clustering; its essence.
- In order to determine what a prediction value is in one record, the user should look for records with similar predictor values in the historical database and use the prediction value from the record that is "nearest" to the unknown record.

Clustering and Nearest Neighbor

- The nearest-neighbor algorithm is basically a refinement of clustering.
- Both techniques use distance in some feature space to create either structure in the data or predictions.
- The nearest-neighbor algorithm is a refinement since part of the algorithm usually is a way of automatically determining the weighting of the importance of the predictors and how the distance will be measured within the feature space.
- Clustering is one special case of this where the importance of each predictor is considered to be equivalent.

Example Nearest Neighbor

- Look at people in a given neighborhood (in this case those people who are, in fact, geographically near to each other).
- In general, they all have somewhat similar incomes.
- Certainly the chances that you have a high income are greater when all your neighbors have incomes over $100,000 than if all your neighbors have incomes of $20,000.
- Within your neighborhood there may still be a wide variety of incomes possible among even your "closest" neighbors, but if you had to predict someone's income based only on knowledge of their neighbors' incomes, your best chance of being right would be to predict the incomes of the neighbors who live closest to the unknown person.

Nearest neighbor for prediction

- One essential element underlying the concept of clustering is that one particular object can be closer to another object than can some third object.
- Most people would agree that an apple is closer to an orange than it is to a tomato and that a Toyota Corolla is closer to a Honda Civic than to a Porsche.
- This definition of nearness that seems to be ubiquitous also allows us to make predictions.
- The nearest-neighbor prediction algorithm, simply stated, is Objects that are "near" to each other will have similar prediction values as well.
- Thus, if you know the prediction value of one of the objects, you can predict it for its nearest neighbors.

The difference between clustering and nearest-neighbor prediction

- The main distinction between clustering and the nearest-neighbor technique is that clustering is an unsupervised learning technique and nearest neighbor is a supervised learning technique.
- Unsupervised learning techniques are unsupervised in the sense that when they are run, there is no particular reason for the creation of the models the way there is for supervised learning techniques that are trying to perform prediction.

The difference between clustering and nearest-neighbor prediction

- In prediction, the patterns that are found in the database and presented in the model are always the most important patterns in the database for performing some particular prediction.
- In clustering there is no particular sense of why certain records are near to each other or why they all fall into the same cluster.
How is the space for clustering and nearest neighbor defined?

- For clustering, the n-dimensional space is usually defined by assigning one predictor to each dimension.
- For the nearest-neighbor algorithm, predictors are also mapped to dimensions, but then those dimensions are literally stretched or compressed according to how important the particular predictor is in making the prediction.
- The stretching of a dimension effectively makes that dimension (and hence predictor) more important than the others in calculating the distance.

How is "nearness" defined?

- The Manhattan distance metric simply adds up the differences between each predictor between the historical record and the record to be predicted.
- The other, called the euclidean distance, calculates distance just the way the Pythagorean theorem calculated the length of the hypotenuse of a triangle (the square of the hypotenuse is equal to the sum of the squares of the other two sides).
- This same calculation is used to calculate the distance between two points in n dimensions by squaring the differences of the predictor values for the two records and then taking the square root of that sum.