

DATA MINING TECHNIQUES TO STUDY VOTING PATTERNS IN THE US

Sikha Bagui*, Dustin Mink, and Patrick Cash

Department of Computer Science, University of West Florida, Pensacola, FL 32514

Email : bagui@uwf.edu

ABSTRACT

This paper presents data mining techniques that can be used to study voting patterns in the United States House of Representatives and shows how the results can be interpreted. We processed the raw data available at <http://clerk.house.gov>, performed t-weight calculations, an attribute relevance study, association rule mining, and decision tree analysis and present and interpret interesting results. WEKA and SQL Server 2005 were used for mining association rules and decision tree analysis.

Keywords: Data mining, Data preprocessing, Attribute relevance study, Association rule mining, Decision tree analysis, Voting patterns.

1 INTRODUCTION

The data mining approach, a relatively new technique, is deployed in large databases to find novel and useful patterns that might otherwise remain unknown. This paper presents a data mining approach to study voting patterns in the US House of Representatives.

The data mining process consists of a series of transformation steps, from data preprocessing to post-processing, as shown in Figure 1 (Tan, Steinbach, & Kumar, 2006). It is the overall process of converting raw data into useful information.

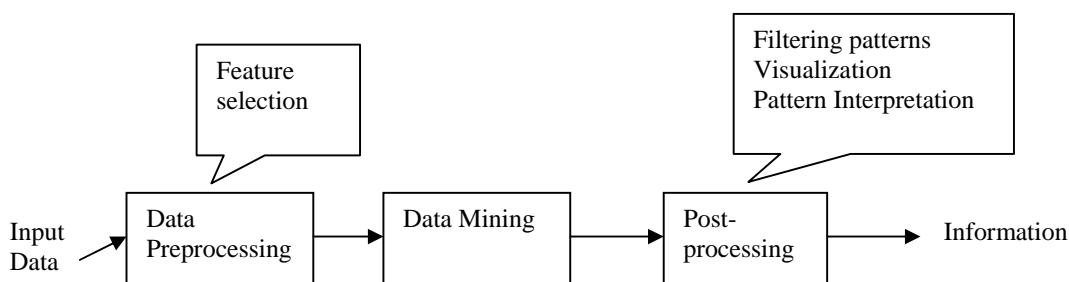


Figure 1. The Data mining process (adopted from Tan, Steinbach, & Kumar, 2006)

Input data can be in various formats (XML data, spreadsheets, relational tables, etc.), and may reside in a centralized location or be distributed over multiple sites. Data preprocessing involves transforming the raw input data into an appropriate format for subsequent data mining. This process involves merging the data from the multiple sources, cleaning the data to remove noise and duplicate observations, and selecting records and features (attributes) that would be relevant to the data mining task at hand. Post-processing ensures that only valid and useful results are incorporated and presented, hence statistical measures are applied to eliminate spurious data mining results. In this paper we present data mining techniques to study voting patterns in the US House of Representatives.

The rest of this paper is organized as follows: section two describes the data; section three explains the data preprocessing efforts; section four presents some exploratory data analysis using t-weights; section five presents an

attribute relevance study; sections six and seven present advanced data mining techniques – association rule mining and decision tree analysis respectively - that we applied to this dataset to determine patterns in this dataset; and section eight presents the conclusions.

For association rule mining and decision tree analysis, Database Management System Software (DBMS), SQL Server 2005, and a machine learning software WEKA were used. SQL Server 2005 is a major DBMS software, and WEKA, which stands for Waikato Environment for Knowledge Analysis, is a collection of machine learning algorithms for solving real world data mining problems. Written in Java, WEKA runs on almost any platform and is available on the web at www.cs.waikato.ac.nz/ml/weka (Witten & Frank, 2000). Results from both SQL Server 2005 and WEKA are presented and discussed.

2 DESCRIPTION OF THE DATA

The United States House of Representatives is one of the two Houses of the Congress of the United States. Each state in the United States is represented in the House proportional to its population, but each state is entitled to at least one Representative. The total number of Representatives is currently 435, each serving a two-year term.

The Office of the Clerk, U.S. House of Representatives website at <http://clerk.house.gov> keeps information on Roll Call Votes by voting issues for both the House and the Senate. This study looks at the pattern of eight pressing issues (voting results) of the 109th Congress, 1st Session (2005), as compiled through the electronic voting machine by the House Tally Clerks under the direction of the Clerk of the House. The voting issues that we looked at are:

- Issue 204 - Stem Cell Research Enhancement Act, available at: <http://clerk.house.gov/evs/2005/roll204.xml>.
- Issue 296 - Proposing an amendment to the Constitution of the United States authorizing the Congress to prohibit the physical desecration of the flag of the United States, available at: <http://clerk.house.gov/evs/2005/roll296.xml>.
- Issue 533 - Personal Responsibility in Food Consumption Act, available at: <http://clerk.house.gov/evs/2005/roll533.xml>.
- Issue 553 - Lawsuit Abuse Reduction Act, available at: <http://clerk.house.gov/evs/2005/roll553.xml>.
- Issue 585 - Secure Access to Justice and Court Protection Act, available at: <http://clerk.house.gov/evs/2005/roll585.xml>.
- Issue 592 - To establish the United States Boxing Commission to protect the general welfare of boxers and to ensure fairness in the sport of professional boxing, available at: <http://clerk.house.gov/evs/2005/roll592.xml>.
- Issue 635 - Pension Protection Act, available at: <http://clerk.house.gov/evs/2005/roll635.xml>.
- Issue 661 - Border Protection, Antiterrorism, and Illegal Immigration Control Act, available at: <http://clerk.house.gov/evs/2005/roll661.xml>.

Each of the above websites contains information on which way (*yea* or *nay*) each member of Congress voted on each of these issues.

3 DATA PREPROCESSING

The data for this study, Roll Call Votes, is available at the different websites given in the previous section (a sample of the Roll Call Votes is presented in Appendix 1). The Roll Call Votes were first saved in XML format (a sample

of which is shown in Appendix 2), then read into a spreadsheet, cleaned up (in terms of removing the extra or unnecessary information), and then integrated into a database.

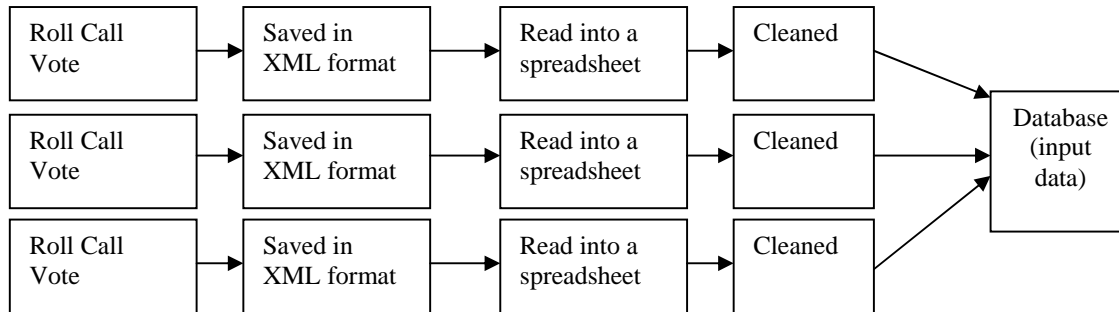


Figure 2. Preprocessing of data

The resulting database contains 435 rows (one for each member of congress) and nine attributes (one for each voting issue and one for the political party of the voting member). In Table 1 we present a sample of the raw data in the database:

Table 1. Sample raw data

Party	V_204	V_296	V_533	V_553	V_585	V_592	V_635	V_661
D	Y	N	N	N	Y	Y	N	N
D	Y	N	N	N	Y	Y	N	N
R	N	Y	Y	Y	Y	N	Y	Y
D	Y	Y	N	N	Y	N	Y	N
D	Y	N	N	N	Y	Y	N	N
R	N	Y	Y	Y	Y	N	Y	Y
R	N	Y	Y	Y	Y	N	Y	Y
...

In the United States, there are two major political parties, Democrat (D) (46.5%) and Republican (R) (52.32%). In this dataset there was only one congressman whose party was “Independent.” Since there was only one member of the Independent party, we eliminated this row from our data set. We also removed any rows (data) where the congressman was not a member of congress at the time the voting issues were selected - this excluded four additional rows of data. So we ended up with 430 rows of data.

Some votes had a “NV” (not voting) recorded for a vote. This meant that the congressman did not submit a vote on a particular issue. No reason was usually given for the no vote (NV). For any issue that had a NV, we replaced it with the majority vote for the party on that particular issue. So, if a Republican did not vote on an issue, but a majority of the other Republicans voted “Y” for that issue, then we replaced the NV with a “Y”.

This made our dataset (or database, as shown in Table 1) ready to be processed or mined. We start our quantitative analysis with an exploratory quantitative analysis tool – t-weight calculations.

4 t-weights

t-weights are an exploratory quantitative data analysis tool that present visualizations of within-class comparisons. For example, in this set of data, t-weights will measure, for each issue, what is the probability that each class will

cast a yes vote or a no vote. So, for each issue, we count the number of yes votes and no votes for each target class, shown by the formula:

$$t_weight = count(q_a) / \sum_{i=1}^n count(q_i)$$

where n is the number of tuples from the target class; $q_1... q_n$ are tuples for the target class and q_a is in $q_1... q_n$. The range for t -weight is [0.0, 1.0] or [0%, 100%]. The t -weight rule is expressed in the form:

$$\forall X, target_class(X) \Rightarrow condition_1(X)[t:w_1] \vee \dots \vee condition_m(X) [t:w_m] \text{ (Han \& Kamber, 2006).}$$

This rule indicates that if X is in the *target class*, there is a probability of w_i that X satisfies condition $_i$, where w_i is the t -weight value for the condition or disjunct i , and is in $\{1, \dots, m\}$ (Han & Kamber, 2006). A rule may not be a sufficient condition of the target class, however, since a tuple satisfying the same condition could also belong to another class.

Using the data from Table 1, we generated t -weights (shown in tables 2 – 9).

Table 2. t -weights – Issue 204

Party	V_204	Count	t-weights
D	N	14	6.93%
D	Y	188	93.07%
R	N	179	78.51%
R	Y	49	21.49%

The t -weights of Table 2 can be converted into logic rules in the form:

Let the target class be Democrats(D). Then the corresponding characteristic rule in logic form is:

Rule 1:

$$\forall X, Party(X) = Democrat \Rightarrow (V_204(X) = "N")[t:6.93\%] \vee (V_204(X) = "Y") [t:93.07\%]$$

This rule says that if X is in the target class, that is, if a member of the House of Representatives is a Democrat, there is a 6.93% probability that this member voted “No” on vote 204, and a 93.07% probability that this member voted “Yes” on Issue 204 (the Stem Cell Research Enhancement Act).

The next rule that can be generated from Table 2 is (here the target class is Republicans):

Rule 2:

$$\forall X, Party(X) = Republican \Rightarrow (V_204(X) = "N")[t:78.51\%] \vee (V_204(X) = "Y") [t:21.49\%]$$

Likewise, this rule says that if X is a republican, there is a 78.51% probability that X voted “No” on vote 204, and a 21.49% probability that X voted “Yes” on Issue 204.

Table 3. t -weights – Issue 296

Party	V_296	Count	t-weights
D	N	125	61.88%
D	Y	77	38.12%
R	N	12	5.26%
R	Y	216	94.74%

The t -weights of Table 3 can be converted to logic rules:

Rule 3:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (\text{V}_{296}(X) = \text{"N"})[t:61.88] \vee (\text{V}_{296}(X) = \text{"Y"}) [t:38.12\%]$$

Rule 4:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (\text{V}_{296}(X) = \text{"N"})[t:5.26\%] \vee (\text{V}_{296}(X) = \text{"Y"}) [t:94.74\%]$$

Table 4. t-weights – Issue 533

Party	V_533	Count	t-weights
D	N	122	60.40%
D	Y	80	39.60%
R	N	1	0.44%
R	Y	227	99.56%

The t-weights of Table 4 can be converted to logic rules:

Rule 5:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (\text{V}_{533}(X) = \text{"N"})[t:60.40] \vee (\text{V}_{533}(X) = \text{"Y"}) [t:39.60\%]$$

Rule 6:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (\text{V}_{533}(X) = \text{"N"})[t:0.44\%] \vee (\text{V}_{533}(X) = \text{"Y"}) [t:99.56\%]$$

Table 5. t-weights – Issue 553

Party	V_553	Count	t-weights
D	N	186	92.08%
D	Y	16	7.92%
R	N	5	2.19%
R	Y	223	97.81%

The t-weights of Table 5 can be converted to logic rules:

Rule 7:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (\text{V}_{553}(X) = \text{"N"})[t:92.08] \vee (\text{V}_{553}(X) = \text{"Y"}) [t:7.92\%]$$

Rule 8:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (\text{V}_{553}(X) = \text{"N"})[t:2.19\%] \vee (\text{V}_{553}(X) = \text{"Y"}) [t:97.81\%]$$

Table 6. t-weights – Issue 585

Party	V_585	Count	t-weights
D	N	44	21.78%
D	Y	158	78.22%
R	N	1	0.44%
R	Y	227	99.56%

The t-weights of Table 6 can be converted to logic rules:

Rule 9:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (\text{V}_{585}(X) = \text{"N"})[t:21.78] \vee (\text{V}_{585}(X) = \text{"Y"}) [t:78.22\%]$$

Rule 10:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (V_{585}(X) = \text{"N"})[t:0.44\%] \vee (V_{585}(X) = \text{"Y"}) [t:99.56\%]$$

Table 7. t-weights – Issue 592

Party	V_592	Count	t-weights
D	N	50	24.75%
D	Y	152	75.25%
R	N	185	81.14%
R	Y	43	18.86%

The t-weights of Table 7 can be converted to logic rules:

Rule 11:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (V_{592}(X) = \text{"N"})[t:24.75] \vee (V_{592}(X) = \text{"Y"}) [t:75.25\%]$$

Rule 12:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (V_{592}(X) = \text{"N"})[t:81.14\%] \vee (V_{592}(X) = \text{"Y"}) [t:18.86\%]$$

Table 8. t-weights – Issue 635

Party	V_635	Count	t-weights
D	N	132	65.35%
D	Y	70	34.65%
R	N	1	0.44%
R	Y	227	99.56%

The t-weights of Table 8 can be converted to logic rules:

Rule 13:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (V_{635}(X) = \text{"N"})[t:65.35] \vee (V_{635}(X) = \text{"Y"}) [t:34.65\%]$$

Rule 14:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (V_{635}(X) = \text{"N"})[t:0.44\%] \vee (V_{635}(X) = \text{"Y"}) [t:99.56\%]$$

Table 9. t-weights – Issue 661

Party	V_661	Count	t-weights
D	N	166	82.18%
D	Y	36	17.82%
R	N	17	7.46%
R	Y	211	92.54%

The t-weights of Table 9 can be converted to logic rules:

Rule 15:

$$\forall X, \text{Party}(X) = \text{Democrat} \Rightarrow (V_{661}(X) = \text{"N"})[t:82.18] \vee (V_{661}(X) = \text{"Y"}) [t:17.82\%]$$

Rule 16:

$$\forall X, \text{Party}(X) = \text{Republican} \Rightarrow (V_{661}(X) = \text{"N"})[t:7.46\%] \vee (V_{661}(X) = \text{"Y"}) [t:92.54\%]$$

4.1 Conclusions for t-weights

From the t-weight rules, we can come up with the following conclusions:

- There is a higher probability of Democrats voting yes and Republicans voting no on stem cell research (Issue 204);
- There is a higher probability of Republicans voting yes and Democrats voting no on the issue of proposing an amendment to the constitution of the United States authorizing the Congress to prohibit the physical desecration of the flag of the United States (Issue 296);
- There is a higher probability of Republicans voting yes and Democrats voting no on the personal responsibility in food consumption act (Issue 533);
- There is a higher probability of Republicans voting yes and Democrats voting no on the Lawsuit Abuse Reduction Act (Issue 553);
- There is a high probability of both the Republicans and Democrats voting yes on the Secure Access to Justice and Court Protection Act (Issue 585);
- There is a higher probability of Democrats voting yes and Republicans voting no on the welfare of boxers act (Issue 592);
- There is a higher probability of Republicans voting yes and Democrats voting no on the Pension Protection Act (Issue 635); and
- There is a higher probability of Republicans voting yes and Democrats voting no on the Border Protection, Antiterrorism, and Illegal Immigration Control Act (Issue 661).

Therefore, from the t-weight conclusions we can see that, except for Issue 585, there is a difference in how the Republicans and Democrats voted.

Though exploratory quantitative generalizations like t-weights give us some information on the data, they are not enough to conclusively say something about the data, so we applied advanced data mining techniques such as association rule mining and decision tree analysis. Association rule mining, one of the most popular data mining techniques (Agrawal, et. al., 1993; Agrawal & Srikant, 1994; Tjioe & Taniar, 2005), helps us discover interesting relationships in large datasets; decision trees help us develop classification rules from the data set. Before applying any of these techniques, however, we thought it would be appropriate to do an attribute relevance analysis to see if there are any weak attributes that should be eliminated from the study. Next we present the attribute relevance analysis.

5 ATTRIBUTE RELEVANCE ANALYSIS

Attribute relevance analysis is used to help identify strong and weak attributes. An attribute is considered strong with respect to a given class if the values of the attribute can be used to distinguish the class from others. The first step in attribute relevance analysis is calculating the information gain.

5.1 Calculating information gain

Let S be a set of training samples, where the class label of each sample is known. Each sample is an example of an instance. Let there be m classes. One attribute, A , is used to determine the class of training samples. Let S contain s_i samples of class C_i , for $i = 1, \dots, m$. An arbitrary sample belongs to class C_i with probability s_i/s , where s is the total number of samples in set S . The expected information needed to classify a sample is (Han & Kamber, 2006):

$$I(S_1, S_2, \dots, S_m) = - \sum_{i=1}^m \frac{s_i}{s} \log_2 \frac{s_i}{s}$$

An attribute A with values $\{a_1, a_2, \dots, a_v\}$ can be used to partition S into the subsets $\{S_1, S_2, \dots, S_v\}$, where S_j contains those samples in S that have value a_j of A . Let S_j contain s_{ij} samples of class C_i . The expected information based on this partition by A is known as the entropy of A . It is the weighted average, shown by (Han & Kamber, 2006):

$$E(A) = \sum_{j=1}^v \frac{s_{1j} + \dots + s_{mj}}{s} I(S_{1j}, \dots, S_{mj})$$

Therefore the information gain obtained by this partitioning on defined by (Han & Kamber, 2006):

$$Gain(A) = I(s_1, s_2, \dots, s_m) - E(A).$$

Using Table 1, we computed the information gain for the target and contrasting class (Democrats and Republicans respectively), as shown in Table 10.

Table 10. Information Gain Calculations

I	S1(D)	S2(R)	I(S1,S2)	E(A)	Gain(A)
I(Total)	202	228	0.997		
I(204Y)	188	49	0.879	0.484	0.204
I(204N)	14	179	0.687	0.308	
I(296Y)	77	216	0.943	0.643	0.147
I(296N)	125	12	0.662	0.211	
I(533Y)	80	227	0.938	0.670	0.174
I(533N)	122	1	0.536	0.153	
I(553Y)	16	223	0.668	0.371	0.361
I(553N)	186	5	0.598	0.265	
I(585Y)	158	227	1.017	0.911	0.049
I(585N)	44	1	0.357	0.037	
I(592Y)	152	43	0.863	0.391	0.123
I(592N)	50	185	0.884	0.483	
I(635Y)	70	227	0.913	0.631	0.199
I(635N)	132	1	0.543	0.168	
I(661Y)	36	211	0.804	0.462	0.232
I(661N)	166	17	0.714	0.304	

5.2 Conclusions from Attribute Relevance Analysis

The information gain for each attribute in order of importance is:

1. Issue 553: 0.361
2. Issue 661: 0.232
3. Issue 204: 0.204
4. Issue 635: 0.199
5. Issue 533: 0.174
6. Issue 296: 0.144
7. Issue 592: 0.123
8. Issue 585: 0.049

Based on these information gain calculations, it would appear that Issue 553 (Personal Responsibility in Food Consumption Act) is the most discriminating dimension. Issue 661 (Border Protection, Antiterrorism, and Illegal Immigration Control Act) is the second most discriminating dimension, and Issue 204 (Stem Cell Research Enhancement Act) is the third most discriminating dimension. Issue 585 (Secure Access to Justice and Court Protection Act), for which both the Democrats and Republicans voted yes as per the t-weight calculations, has the lowest information gain (that is, it is the least discriminating attribute) as per the attribute relevance analysis too. The other issues are ranked as indicated. We, however, decided to keep all our attributes for future analysis. Next we present association rule mining.

6 ASSOCIATION RULE MINING

Association rule mining techniques are used to discover interesting associations between attributes in a database. The classical definition of association rules, as presented in Agrawal, Imielinski, & Swami (1993) and Han & Kamber, (2006)) is: Let $\{t_1, t_2 \dots t_N\}$ be a set of transactions, and let I be a set of items, $I = \{i_1, i_2 \dots i_M\}$. Let D , the task-relevant data, be a set of transactions where each transaction T is a set of items such that $T \subseteq I$. Let X be a set of items. A transaction T is said to contain X if and only if $X \subseteq T$. An association rule is an implication of the form $X \Rightarrow Y$, where $X, Y \subset I$, and X and Y are disjoint itemsets, i.e. $X \cap Y = \emptyset$. Below we present the algorithm used to mine association rules.

6.1 Algorithm to mine association rules

The association mining rule algorithm, known as the Apriori algorithm, is an algorithm that finds frequent itemsets using an iterative approach based on candidate generation. Below we present the pseudocode for the Apriori algorithm, as shown in (Han & Kamber, 2006):

```

L1 := {frequent 1-itemsets} D;
  for (k=2; Lk-1 ≠ ∅; k++)
    Ck = apriori_gen(Lk-1, min_sup);
    for each transaction t ∈ D { //scan D for counts
      Ct = subset(Ck, t); // get the subsets of t that are candidates
      for each candidate c ∈ Ct
        c.count++;
    }
    Lk = { c ∈ Ck | c.count ≥ min_sup }
  }
return L = ∪k Lk;

procedure apriori_gen(Lk-1; frequent(k-1)-itemsets; min_sup: minimum support threshold)
  for each itemset l1 ∈ Lk-1
    for each itemset l2 ∈ Lk-1
      if (l1[1] = l2[1]) ∧ (l1[2] = l2[2]) ∧ ... ∧ (l1[k-2] = l2[k-2]) ∧ (l1[k-1] = l2[k-1]) then {
        c = l1 l2; // join step: generates candidates
        if has_infrequent_subset(c, Lk-1) then
          delete c; //prune step: remove unfruitful candidate
        else add c to Ck;
      }
  }
return Ck;

procedure has_infrequent_subset(c:candidate k-itemset; Lk-1: frequent (k-1)-itemsets);
  //use prior knowledge
  for each (k-1) – subset s of c
    if s ∉ Lk-1 then
      return TRUE;
  return FALSE;

```

This Apriori algorithm employs an iterative approach, where k -itemsets are used to explore $(k+1)$ – itemsets. First, the set of frequent 1-itemsets is found, denoted by L_1 . All these frequent 1-itemsets have to have *support* above a user-specified minimum. The frequent 1-itemsets are generated by counting item occurrences and then using those that turn out to be frequent after computing their support.

L_1 is then used to find L_2 , the set of frequent 2-itemsets, which in turn is used to find L_3 , and so on until no more frequent k -itemsets can be found. The size of the itemsets is incremented by one at each iteration, and the finding of each L_k requires one full scan of the database. This phase stops when there are no additional frequent itemsets.

The apriori_gen procedure performs two steps – a join and a prune. In the join part, L_{k-1} is joined with L_{k-1} to generate potential candidates. The prune portion employs the Apriori property to remove candidates that have a subset that is not frequent. The test for infrequent subsets is shown in procedure has_infrequent_subset (Han & Kamber, 2006).

Association rules can have one or several output attributes, so association rules help us predict any attribute or a combination of attributes. This is a popular technique for data analysis as all possible combinations of potentially interesting groupings in the data can be explored. Because all possible groupings are derived from association rules, a large number of association rules can be derived from any data set. Hence, interest in an association rule is restricted to those rules that apply to a reasonably large number of instances and have a reasonably high accuracy on the instances that they apply to. We used WEKA and SQL Server 2005 to generate association rules.

6.2 Generating Association rules using WEKA

WEKA generates association rules that have one or several output attributes. The strength of an association rule in WEKA is measured in terms of the rule's statistical significance, known as *support* and *confidence*. Support s is the percentage of transactions in D that contain $X \cup Y$, that is, the probability, $P(X \cup Y)$. Confidence c is the percentage of transactions in D containing X that also contain Y , that is the conditional probability, $P(X|Y)$ (Han & Kamber, 2006).

Therefore, our task at hand for this data set was to find all the association rules having *support* \geq *min_Support* and *confidence* \geq *min_Confidence*. We started with a support of 90% and confidence of 90%, but got no rules. We kept changing (lowering) the support, but got no rules until we input a support of 55% (and the confidence was 90%), for which the rules are presented below:

=== Run information ===

Scheme: weka.associations.Apriori -N 20 -T 0 -C 0.9 -D 0.05 -U 1.0 -M 0.5 -S -1.0

Apriori
=====

Minimum support: 0.55
Minimum metric <confidence>: 0.9
Number of cycles performed: 9

Best rules found:

1. 296=y 533=y 635=y 243 ==> 585=y 243 conf:(1)
2. 533=y 661=y 239 ==> 585=y 239 conf:(1)
3. 533=y 553=y 237 ==> 585=y 237 conf:(1)
4. 533=y 635=y 269 ==> 585=y 268 conf:(1)
5. 296=y 533=y 264 ==> 585=y 263 conf:(1)
6. 661=y 247 ==> 585=y 246 conf:(1)
7. 553=y 239 ==> 585=y 238 conf:(1)
8. 553=y 585=y 238 ==> 533=y 237 conf:(1)
9. 296=y 635=y 254 ==> 585=y 252 conf:(0.99)
10. 553=y 239 ==> 533=y 585=y 237 conf:(0.99)
11. 553=y 239 ==> 533=y 237 conf:(0.99)
12. 533=y 307 ==> 585=y 303 conf:(0.99)
13. 296=y 293 ==> 585=y 287 conf:(0.98)
14. 585=y 661=y 246 ==> 533=y 239 conf:(0.97)
15. 635=y 297 ==> 585=y 288 conf:(0.97)
16. 661=y 247 ==> 533=y 585=y 239 conf:(0.97)
17. 661=y 247 ==> 533=y 239 conf:(0.97)
18. 296=y 585=y 635=y 252 ==> 533=y 243 conf:(0.96)
19. 296=y 635=y 254 ==> 533=y 585=y 243 conf:(0.96)
20. 296=y 635=y 254 ==> 533=y 243 conf:(0.96)

6.3 Discussion of the association rules generated with WEKA

WEKA generates all possible groupings of associations. Below we present some of the association rules that can be generated from the WEKA output:

From Rule 1:

Vote “yes” on Issue 296 and Vote “yes” on Issue 533 and Vote “yes” on Issue 635 is associated with Vote “yes” on Issue 585.

From Rule 2:

Vote “yes” on Issue 533 and Vote “yes” on Issue 661 is associated with Vote “yes” on Issue 585.

From Rule 3:

Vote “yes” on Issue 533 and Vote “yes” on Issue 553 is associated with Vote “yes” on Issue 585.

WEKA’s results grouped the issues, but did not present any associations with the political parties. Next we generated association rules in SQL Server 2005 to see if we could associate the issues to the political parties.

6.4 Association rules generated using SQL Server 2005

In SQL Server 2005, the strength of an association rule is measured in terms of the rule’s statistical significance, the probability, and importance. We used a minimum probability of 0.85 and minimum importance of 0.90, and SQL Server 2005 generated the rules shown in figure 3:

Pr...	Importance	Rule
1.000	0.989	553 = N, 204 = Y -> Party = D
0.974	1.138	553 = N -> Party = D
0.946	1.039	553 = Y, 296 = Y -> Party = R
0.945	1.423	553 = Y, 635 = Y -> Party = R
0.945	0.942	553 = Y, 661 = Y -> Party = R
0.937	1.415	553 = Y, 533 = Y -> Party = R
0.933	1.411	553 = Y, 585 = Y -> Party = R
0.933	1.476	553 = Y -> Party = R
0.890	0.961	661 = Y, 635 = Y -> Party = R
0.879	0.949	661 = Y, 533 = Y -> Party = R
0.854	0.921	661 = Y, 585 = Y -> Party = R
0.854	0.942	661 = Y -> Party = R

Figure 3. Association rules generated using SQL Server 2005

Inferences made from association rules do not necessarily imply causality, but suggest a strong co-occurrence relationship between the antecedent and consequent of the rule. Below we present some of the associations that can be implied from the SQL Server 2005 output generated above:

Association rule 1:

Vote “no” on Issue 553 (Lawsuit Abuse Reduction Act) and vote “yes” on Issue 204 (Stem Cell Research Enhancement Act) is associated with Democrats.

And this has a probability of 100%.

Association rule 2:

Vote “no” on Issue 553 (Lawsuit Abuse Reduction Act) is associated with Democrats.

Association rule 3:

Vote “yes” on Issue 553 (Lawsuit Abuse Reduction Act) and vote “yes” on Issue 296 (Proposing amendment to the Constitution of the United States authorizing Congress to prohibit physical desecration of the flag of the United States) is associated with Republicans.

Association rule 4:

Vote “yes” on Issue 553 (Lawsuit Abuse Reduction Act) and vote “yes” on Issue 635 (Pension Protection Act) is associated with Republicans.

6.5 Discussion of association rules presented using SQL Server 2005

Because we could control the class attribute in generating the association rules in SQL Server 2005, this output helped us find some strong associations between the issues being studied and the political party (Democrat or Republican). From the above results we can clearly see: a high probability associated with a no vote on issue 553 (Lawsuit Abuse Reduction Act) and Democrats; a high probability associated a “yes” vote on Issue 553 (Lawsuit Abuse Reduction Act) and Issue 296 (Proposing amendment to the Constitution of the United States authorizing Congress to prohibit physical desecration of the flag of the United States) is associated with Republicans, etc. These results are also in line with the t-weight results.

7 DECISION TREE ANALYSIS

A decision tree is a structure where non-terminal nodes represent tests on one or more attributes and terminal nodes reflect decision outcomes. Decision trees are a useful data analysis tool as they are easy to understand and can be easily transformed into rules. Decision trees are constructed using only those attributes best able to differentiate concepts. The main goal in a decision tree algorithm is to minimize the number of tree levels and tree nodes, thereby maximizing data generalization.

The C4.5 (Quinlan, 1993) decision tree algorithm uses a measure taken from information theory to help with the attribute selection process. At each choice point in the tree, C4.5 computes the gain ratio for all available attributes. The attribute with the largest value for this ratio is selected to split the data. This attribute becomes the “test” or “decision” attribute. A branch is then created for each known value of the test attribute, and the samples partitioned accordingly. The algorithm uses this same process recursively to form a decision tree. There are two possibilities for terminating the path of a tree: first, if the instances following a given branch satisfy a predetermined criterion, such as a minimum training set classification accuracy, the branch becomes a terminal path. A second possibility for terminating a path of the tree is the lack of an attribute for continuing the tree splitting process. An obvious termination criterion is that all instances following a specific path must be from the same class (Roiger & Geatz, 2003).

7.1 Decision Tree Generated Using SQL Server 2005

In Figure 4 we present the decision tree generated from SQL Server 2005:

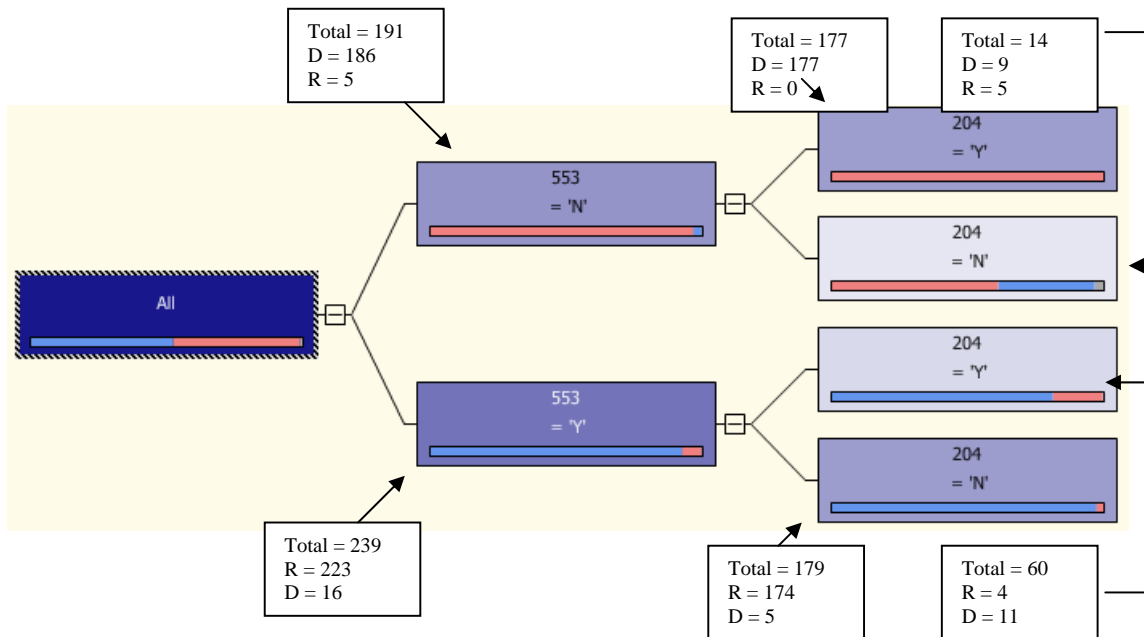


Figure 4. Decision tree generated using SQL Server 2005

7.1.1 Discussion of results of decision tree generated using SQL Server 2005

According to this decision tree, most Democrats voted “no” on Issue 553, and Republicans voted “yes” on Issue 553. Of those that voted no, most Democrats voted “yes” on Issue 204, and all the Republicans voted “no” on Issue 204. Of those that voted yes on Issue 553, most Republicans voted “no” on Issue 204 and most Democrats voted “yes” on Issue 204. Again, these results are in line with the exploratory t-weight analysis results and, as per the attribute relevance study, the highest information gain was Issue 553, and this is also at the root of the decision tree.

We found a few limitations with the decision tree generated in Since SQL Server 2005: (i) it does not give us an opportunity to control the confidence factor; (ii) it does not give us an opportunity to control the size of the training samples; (iii) we could not see what the error rate was; (iv) it was not generating more levels, so we were not able to determine how the other issues played in to the decision between the democrats and republicans. Hence, we decided to use WEKA to generate our decision tree.

7.2 Decision Tree Generated Using WEKA

The decision tree algorithm that we used in WEKA, J48, gives us an opportunity to control the confidence factor and training sample size (controlled by the cross-validation option). Our objective is to get a decision tree that minimizes the expected error rate, with the highest amount of correctly classified instances.

After running WEKA with different confidence values, a confidence of 98% and 2-fold cross validation seemed to give us the highest amount of correct classification; hence the decision tree was generated with 98% confidence and 2-fold cross validation (party was used as our class variable) – this decision tree is shown in Figure 5.

In WEKA, the confidence factor is used to address the issue of tree pruning. When a decision tree is being built, many of the branches will reflect anomalies due to noise or outliers in the training data. Tree pruning uses statistical measures to remove these noise and outlier branches, allowing for the confidence factor. This means that our data set did not have much noise or outlier cases, so there was not much to prune faster classification and improvement in the ability of the tree to correctly classify independent test data (Han & Kamber, 2006). A smaller confidence factor will incur more pruning, so for example if a 98% confidence factor is used, our tree will incur less pruning. We ran WEKA with a very wide range of confidence factors, but the results were not reacting to.

Cross validation determines the amount of data to be used for reduced-error pruning (the training and the test set). If a 2-fold is used, one fold (of the data) is used for pruning and the other fold (rest of the data) is used for growing the tree. Our data set gave the same results for 2-10 fold.

Our highest percentage (95.1163%) of correctly classified instances (and minimum error) was produced with almost any confidence factor and cross validation anywhere from 2-10. Below is the output and decision tree produced with these results:

WEKA GENERATED OUTPUT:

=== Run information ===
 === Classifier model (full training set) ===

J48 pruned tree

```

    553 = n
    | 204 = y: Democrat (177.0)
    | 204 = n
    | | 533 = n: Democrat (6.0)
    | | 533 = y
    | | | 592 = y: Democrat (5.0/2.0)
    | | | 592 = n: Republican (3.0)
    553 = y
    | 635 = n: Democrat (4.0/1.0)
    | 635 = y: Republican (235.0/13.0)
    
```

=== Stratified cross-validation ===
 === Summary ===

Correctly Classified Instances	409	95.1163 %
Incorrectly Classified Instances	21	4.8837 %

=== Confusion Matrix ===

```

    a  b  <-- classified as
    185 17 | a = Democrat
    4  224 | b = Republican
    
```

Figure 5 shows the decision tree generated by WEKA.

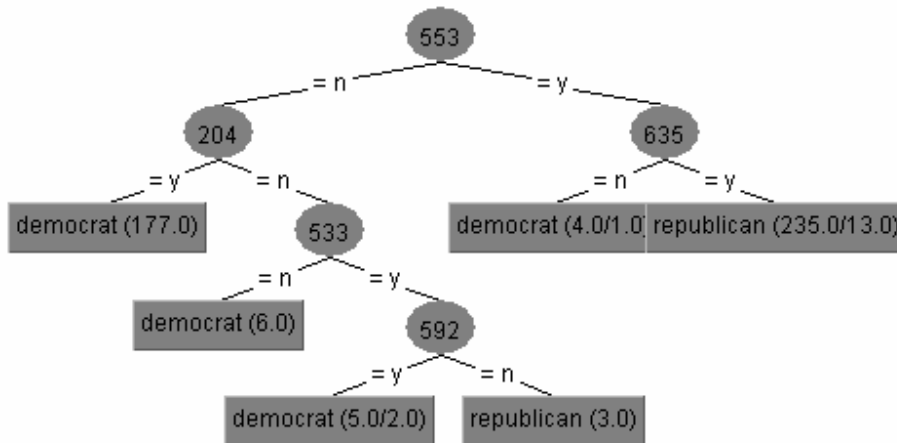


Figure 5. WEKA generated decision tree

7.2.1 Discussion of the results of the decision tree generated using WEKA

The order of the attributes in the decision tree is based on information gain. For example, as we saw from the attribute relevance analysis, the attribute with the highest information gain is Issue 553; hence this issue is at the root of the tree. Therefore, Issue 553 appears to be the most discriminating issue between Republicans and Democrats. For those members that voted no on Issue 553, Issue 204 was the next most discriminating attribute. For those that voted yes on Issue 553, Issue 635 was the next most discriminating attribute. From the attribute relevance analysis we can also see that Issues 204 and 635 were the next most discriminating attributes respectively, and so on. Below we present the classification rules that can be developed from this decision tree.

7.2.1.1 Classification rules developed from decision tree generated from WEKA

Knowledge from a decision tree can be extracted and presented in the form of classification IF-THEN rules. One rule is created for each path from the root to a leaf node. Each attribute-value pair along a given path forms a conjunction in the rule antecedent (“IF” part). The leaf node holds the class prediction, forming the rule consequent (“THEN”) part (Han & Kamber, 2006). Now we present rules that can be developed from the above decision tree:

*IF Issue 553 has vote YES and Issue 635 has vote YES
THEN Party is REPUBLICAN*

*IF Issue 553 has vote NO and Issue 204 has vote YES
THEN Party is DEMOCRAT*

These above results are also in line with the results obtained from association rule mining.

*IF Issue 553 has vote NO and Issue 204 has vote NO and Issue 553 has vote NO
THEN Party is DEMOCRAT*

*IF Issue 553 has vote NO and Issue 204 has vote NO and Issue 553 has vote YES and Issue 592 has vote YES
THEN Party is DEMOCRAT*

*IF Issue 553 has vote NO and Issue 204 has vote NO and Issue 553 has vote YES and Issue 592 has vote NO
THEN Party is REPUBLICAN*

8 CONCLUSIONS

In this paper we presented techniques that can be used to study or mine voting patterns in the US House of Representatives. We have shown the whole data mining processing – from processing input data to preprocessing to attribute relevance analysis to the use of advanced data mining techniques like association rule mining and decision tree generation and analysis to presenting information (in the form of rules) and conclusions.

The exploratory data mining techniques, t-weights, gave us a picture of what percentage of each party voted on a particular issue. The attribute relevance analysis showed which issues were the most discriminating (in order); WEKA’s association mining results showed us which issues can be grouped together; SQL Server 2005’s association mining results showed us which issues were associated to which political parties, and the decision tree results helped us classify the political party of a member based on what they voted for. From this preliminary study of voting patterns, we can see that there is quite a bit of difference in how the Democrats and Republicans vote, and we got some interesting results, presented and discussed in each of the sections studied. Having obtained encouraging results so far, our future plan is to work in the direction of analyzing the rest of the issues (and there are many) and develop an overall pattern as to how the Democrats and Republicans vote in the US House of Representatives.

9 REFERENCES

Agrawal, R., Imielinski, T., & Swami, A. (1993) Mining association rules between sets of items in large databases. *ACM SIGMOD Conference*, pp. 207-216.

Agrawal, R. & Srikant, R. (1994) Fast Algorithms for Mining Association Rules in Large Databases. *Proc. 20th Int'l Conf. Very Large Data Bases*, pp. 478-499.

Han, J. & Kamber, M. (2006) *Data Mining: Concepts and Techniques*. USA: Morgan Kaufmann Publishers.

<http://clerk.house.gov>

<http://clerk.house.gov/evs/2005/roll204.xml>.

<http://clerk.house.gov/evs/2005/roll296.xml>.

<http://clerk.house.gov/evs/2005/roll533.xml>.

<http://clerk.house.gov/evs/2005/roll553.xml>.

<http://clerk.house.gov/evs/2005/roll585.xml>.

<http://clerk.house.gov/evs/2005/roll592.xml>.

<http://clerk.house.gov/evs/2005/roll635.xml>.

<http://clerk.house.gov/evs/2005/roll661.xml>.

Quinlan, J.R. (1993) *Programs for Machine Learning*. San Mateo, CA: Morgan Kaufman.

Roiger, R. & Geatz, M. (2003) *Data Mining: A Tutorial-Based Primer*. Addison Wesley.

Srikant, R. & Agrawal, R. (1997) Mining Generalized Association Rules. *Future Generation Computer Systems*, 13:2-3

Tan, P-N, SteinBach, M., & Kumar, V. (2006) *Introduction to Data Mining*. Addison Wesley.

Tjioe, H. C. & Taniar, D. (2005) Mining Association Rules in Data Warehouses. *International Journal of Data Warehousing and Mining*, 1(3), pp.28 – 62.

Witten, I.H. & Frank, E. (2000) *Data Mining: Practical Machine Learning Tools and Techniques with Java Implementations*. Morgan Kaufmann.

www.cs.waikato.ac.nz/ml/weka.

10 APPENDIX 1 – EXAMPLE OF ROLL CALL VOTES

FINAL VOTE RESULTS FOR ROLL CALL 204

(Republicans in roman; Democrats in *italic*; Independents underlined)

H R 810 YEA-AND-NAY 24-May-2005 6:07 PM

QUESTION: On Passage

BILL TITLE: Stem Cell Research Enhancement Act

	YEAS	NAYS	PRES	NV
REPUBLICAN	50	180		1
DEMOCRATIC	187	14		1
INDEPENDENT	1			
TOTALS	238	194		2

--- YEAS 238 ---

Abercrombie
Ackerman
Allen
Andrews
Baca
Baird

.
.
.

--- NAYS 194 ---

Aderholt
Akin
Alexander
Bachus
Baker
Barrett (SC)

.
.
.

--- NOT VOTING 2 ---

Hastings (WA)	Millender-McDonald
---------------	--------------------

11 APPENDIX 2 – ROLL CALL VOTES IN XML

```
<?xml version="1.0" encoding="UTF-8"?>
<!DOCTYPE rollcall-vote PUBLIC "-//US Congress//DTDs/vote v1.0 20031119 //EN" "http://clerk.house.gov/evs/vote.dtd">
<?xml-stylesheet type="text/xsl" href="http://clerk.house.gov/evs/vote.xsl"?>
<rollcall-vote>
<vote-metadata>
<congress>109</congress>
<session>1st</session>
<chamber>U.S. House of Representatives</chamber>
<rollcall-num>204</rollcall-num>
<legis-num>H R 810</legis-num>
<vote-question>On Passage</vote-question>
<vote-type>YEA-AND-NAY</vote-type>
<vote-result>Passed</vote-result>
<action-date>24-May-2005</action-date>
<action-time time-etz="18:07">6:07 PM</action-time>
<vote-desc>Stem Cell Research Enhancement Act</vote-desc>
<vote-totals>
<totals-by-party-header>
<party-header>Party</party-header>
<yea-header>Yeas</yea-header>
```

```
<nay-header>Nays</nay-header>
<present-header>Answered "Present"</present-header>
<not-voting-header>Not Voting</not-voting-header>
</totals-by-party-header>
.
.
.
</totals-by-vote>
</vote-totals>
</vote-metadata>
<vote-data>
<recorded-vote><legislator name-id="A000014" sort-field="Abercrombie" unaccented-name="Abercrombie" party="D" state="HI"
role="legislator">Abercrombie</legislator><vote>Yea</vote></recorded-vote>
<recorded-vote><legislator name-id="A000022" sort-field="Ackerman" unaccented-name="Ackerman" party="D" state="NY"
role="legislator">Ackerman</legislator><vote>Yea</vote></recorded-vote>
<recorded-vote><legislator name-id="A000055" sort-field="Aderholt" unaccented-name="Aderholt" party="R" state="AL"
role="legislator">Aderholt</legislator><vote>Nay</vote></recorded-vote>
<recorded-vote><legislator name-id="A000358" sort-field="Akin" unaccented-name="Akin" party="R" state="MO"
role="legislator">Akin</legislator><vote>Nay</vote></recorded-vote>
.
.
.
```