CSCI577 Final

1 DATA TYPES

2 MATH

$$log_b a = \frac{log_x a}{log_x b}$$

3 CLASSIFICATION

Definition. Constructing a method of classifying new instances using information in a training set

- Naive Bayes (Conditional Probabilities)
- Decision Trees
- Logistic Regression
- Neural Networks

3.1 Naïve Bayes

Bayes Theorem:

$$P(A|B) = \frac{P(A,B)}{P(B)} = \frac{P(B|A) \times P(A}{P(B)}$$

Calculation. Probablity of a class given attributes is a product of probability of that class overall, with the sum product of each individual attribute given the class:

$$P(c_i|v) = P(c_i)\prod_{j=1}^{n} P(a_j = v_j|\text{class} = c_i)$$

3.2 TDIDT

Definition. Top-Down Induction of Decision Trees

Algorithm. Until no more splitting is possible:

- IF all the instances in the training set belong to the same class THEN return the value of the class
- ELSE
  1. (a) Select an attribute A to split on
  2. (b) Sort the instances in the training set into subsets, one for each value of attribute A
  3. (c) Return a tree with one branch for each non-empty subset
     - Each branch having a descendant subtree or a class value produced by applying the algorithm recursively

3.3 Adequacy

Definition. No two instances with the same values of all the attributes may belong to different classes. Naive bayes can still be used when this doesn't obtain, as it will still be able to obtain the probabilities of each class. kNN can still be used, as long as then multiple datapoints in the same location in euclidean space would still function as expected.

3.4 Overfitting

Understand the concept of overfitting and be able to tell how you would know that a classification system overfit

Definition. If classifier generates a decision tree (or other mechanism) too well adapted to the training set Performs well on training set, not well on other data. Some overfitting inevitable.

Remedy:

- Adjust a decision tree while it is being generated: Pre-pruning
- Modify the tree after creation: Post-pruning

3.4.1 Clashes. Two (or more) instances of a training set have identical attribute values, but different classification. Especially a problem for TDIDT's 'Adequacy condition'.

Stems from.

- Classification incorrectly recorded
- Recorded attributes insufficient - Would need more attributes, normally impossible

Solutions.

- Discard the branch to the clashing node from the node above
- Of clashing instances, assume the majority label

3.4.2 Prepruning. When pre-pruning, may reduce accuracy on training set but may be better on test set (and subsequent) data than unpruned classifier.

1. test whether a termination condition applies.
   - If so, current subset is treated as a 'clash set'
   - Resolve by 'delete branch,' 'majority voting,' etc.
2. two methods methods
   - Size Cuttoff – prune of subset has fewer than X instances
   - Maximum depth: prune if length of branch exceed Y

3.4.3 PostPruning.

1. look for a non-leaf nodes that have descendants of length 1.
2. In this tree, only node G and D are candidates for pruning (consolidation).

3.5 Discretizing

3.5.1 Equal Width Intervals.

3.5.2 Pseudo Attributes.

3.5.3 Processing Sorted Instance Table.

3.5.4 ChiMerge.

Rationalization. Initially, each distinct value of a numerical attribute A is considered to be one interval. χ^2 tests are performed for every pair of adjacent intervals. Adjacent intervals with the least χ^2 values are merged together, because χ^2 low values for a pair indicates similar class distributions. This merging process proceeds recursively until a predefined stopping criterion is met. For two adjacent intervals, if χ^2 test concludes that the class is independent intervals should be merged. If χ^2 test concludes that they are not independent, i.e. the difference in relative class frequency is statistically significant, the two intervals should remain separate.

Calculation. To calculate expected value for any combination of row and class:

1. Take the product of the corresponding row sum and column sum
2. Divided by the grand total of the observed values for the two rows.

Then:

1. Using observed and expected values, calculate, for each of the cells: $\frac{(O-E)^2}{E}$
2. Sum each cell's χ^2

When exceeds χ^2 threshold, hypothesis is rejected. Small value for supports hypothesis. Important adjustment, when $E < 0.5$ replace it with 0.5.

1. Select the smallest value
2. Compare it to the threshold
3. If it falls below the threshold, merge it with the row immediately below it
4. recalculate χ^2, Only need to do this for rows adjacent to the recently merged one.

Large numbers of intervals does little to solve the problem of discretization. Just one interval cannot contribute to a decision making process. Modify significance level hypothesis of independence must pass, triggering interval merge. Set a minimum and a maximum number of intervals

Table 1: data types

Variable type	Description	Examples
Categorical		
Nominal (unordered)	Gives only qualitative information	Names, occupations, nationalities, sex, religion
Ordinal (ordered)	Ranking or order is important	Social status, economic class
Numeric		
Interval	Distance between values has meaning (discrete or continuous)	Year, temperature
Ratio	Ratio of two values has meaning	Wealth, age, prices, wages

3.6 Entropy

Definition. Entropy is the measure of the presence of there being more than one possible classification. Used for splitting attributes in decision trees Entropy minimizes the complexity, number of branches, in the decision tree. No guarantee that using entropy will always lead to a small decision Tree Used for feature reduction: Calculate the value of information gain for each attribute in the original dataset. Discard all attributes that do not meet a specified criterion. Pass the revised dataset to the preferred classification algorithm

Entropy has bias towards selecting attributes with a large number of values

Calculation. To decide if you split on an attribute:

(1) find the entropy of the data in each of the branches after the split
(2) then take the average of those and use it to find information gain.
(3) The attribute split with the highest information gain (lowest entropy) is selected.

- Entropy is always positive or zero
- Entropy is zero when $p_i = 1$, aka when all instances have the same class
- Entropy is at its max value for the # of classes when all classes are evenly distributed

If there are classes, we can denote the proportion of instances with classification i by p_i for $i = 1toK$. $p_i = \frac{\text{instances of class} i}{\text{total number of instances}}$

$$\text{Entropy} = E = -\sum_{i=1}^{K} p_i log_2 p_i$$

where K = non-empty classes and $p_i = \frac{|i|}{N}$, instances in class i over total number of instances N.

3.7 GINI

Calculation.

(1) For each non-empty column, form the sum of the squares of the values in the body of the table and divide by the column sum.
(2) Add the values obtained for all the columns and divide by N (the number of instances).
(3) Subtract the total from 1.

3.8 Information Gain

Definition. The difference between the entropy before and after splitting on a given attribute in a decision tree. Maximizing information gain is the same as minimizing E_{new}.

Calculation.

$$\text{Information Gain} = E_{\text{start}} - E_{\text{new}}$$

Starting node:

$$\begin{aligned} E_{\text{start}} = &-\frac{4}{24} log_2 \frac{4}{24} \\ &-\frac{5}{24} log_2 \frac{5}{24} \\ &-\frac{15}{24} log_2 \frac{15}{24} \end{aligned} \quad (1)$$

After spliting on attribute:

$$\begin{aligned} E_{\text{new}} = &\frac{8}{24} E_1 \\ &+\frac{8}{24} E_2 \\ &+\frac{8}{24} E_3 \end{aligned} \quad (2)$$

Uses.

4 CLUSTERING

Definition. Grouping data into seperate groups. Use distance metric between two datapoints. Groups should be distinct from another and composed of items similar to one another, and different from items in other groups.

4.1 Nearest Neighbors

Mainly used when all attribute values are continuous

General strategy:

(1) Find the k training instances that are closest to the unseen instance.
(2) Take the most commonly occurring classification for these instances.

- KMeans
- DBSCAN

5 SEQUENCE MINING

TODO

Definition. Finding meaningful, recurring sequences of events

- A sequence is an ordered list of elements (transactions):
$$s =< e_1 e_2 e_3 >$$
- Each element contains a collection of events (items):
$$e_i = i_1 i_2 i_3 \cdots i_k$$
- Each element is attributed to a specific time or location.
- Length of a sequence, $|s|$, is given by the number of elements of the sequence.
- A k-sequence is a sequence that contains k events (items)

Contains. A sequence $< a_1 a_2 \cdots a_n >$ is contained in another sequence $< b_1 b_2 \cdots b_m >; (m \geq n)$ if there exist integers $i_1 < i_2 < \cdots < i_n$ such that $a_1 \subseteq b_{i1} a_2 \subseteq b_{i2} \cdots a_n \subseteq b_{in}$.

Support. The support of a subsequence w is defined as the fraction of data sequences that contain w. A sequential pattern is a frequent subsequence (i.e., a subsequence where support ≥ minsup)

5.1 Generalized Sequential Pattern

(1) Make the first pass over the sequence database D to yield all the 1-element frequent sequences
(2) Repeat until no new frequent sequences are found:
  (a) Candidate Generation: Merge pairs of frequent subsequences found in the $(k-1)$'th pass to generate candidate sequences that contain k items
  (b) Initial Pruning: Prune if it is not the case that all of the $k-1$ subsequences of a k sequence are frequent
  (c) Support Counting: Make a new pass over the sequence database D to find the support for these candidate sequences
  (d) Candidate Elimination: Eliminate candidate k-sequences whose actual support is less than minsup

5.2 Counting Methods

- COBJ: One occurrence per object
- CWIN: One occurrence per sliding window
- CMINWIN: Number of minimal windows of occurrence
- CDIST 0: Distinct occurrences with possibility of event-timestamp overlap
- CDIST: Distinct occurrences with no event- timestamp overlap allowed

6 ASSOCIATION RULE ANALYSIS

Definition. Given a collection of collections (database of transactions of food items), find items with high co-occurance.

Let m be the number possible items that can be bought. Let I denote the set of all possible items. Possible itemsets: $s^{|I|}$ An itemset S matches a transaction T (itself an itemset) if $S \subset T$.

6.1 Support

Definition. $support(S)$: proportion of itemsets matched by S. Proportion of transactions that contain all the items in S. Frequency with which the items in S occur together in the database.

Calculation.

$$\text{support}(S) = \frac{count(S)}{n}$$

where n is the number of transactions in the database.

6.2 APRIORI

Pseudo-code.

```
(1) Create $L_1$ = set of supported itemsets of cardinality one.
(2) Set $k = 2$.
(3) while ($L_{k-1} \neq$).
    (a) Create $C_k$ from $L_{k-1}$.
    (b) Prune all the itemsets in $C_k$ that are not supported, to create $L_K$.
    (c) Increase $k$ by 1.
(4) The set of all supported itemsets is $L_1 \cup L_2 \cup \cdots \cup L_k$.
```

To start the process we construct C_1.

(1) Set of all itemsets comprising just a single item,
(2) Make a pass through the database counting the number of transactions that match each of these itemsets.
(3) Divide these counts by the number of transactions in the database
(4) Checking for minsup each single-element itemset.
(5) Discard all those with support $< minsup$ to yield L_k.
(6) Continue until is empty.

6.3 Confidence

Calculation. Confidence of a rule can be calculated either by

$$Confidence(L \to R) = \frac{count(L \cup R)}{count(L)}$$

or

$$Confidence(L \to R) = \frac{support(L \cup R)}{support(L)}$$

Reject rules where

$$support < minsup \approx 0.01 = 1\%$$

Also called a frequent|large|supported itemset.

Reject rules where

$$confidence < minconf \approx 0.8 = 80\%$$

Uses.

6.4 Lift

Definition. Lift measures how many more times the items in and occur together than would be expected if they were statistically independent. Although lift is a useful measure of interestingness it is not always the best one to use. In some cases a rule with higher support and lower lift can be more interesting than one with lower support and higher lift because it applies to more cases.

Calculation.

$$\begin{aligned} \text{Lift}(L \to R) &= \frac{\text{count}(L \cup R)}{\text{count}(L) \times \text{support}(R)} \\ &= \frac{\text{support}(L \cup R)}{\text{support}(L) \times \text{support}(R)} \\ &= \frac{\text{confidence}(L \to R)}{\text{support}(R)} \\ &= \frac{N \times \text{confidence}(L \to R)}{\text{count}(R)} \\ &= \frac{N \times \text{confidence}(R \to L)}{\text{count}(R)} \\ &= \text{Lift}(R \to L) \end{aligned} \tag{3}$$

6.5 Leverage

Calculation.

$$\text{leverage}(L \to R) = \text{support}(L \cup R) - \text{support}(L) \times \text{support}(R)$$

6.6 Frequent Itemsets

(1) Find itemsets of size k made from 2 supported itemsets of size $k-1$
(2) For each new itemset:
  (a) check if every sub-itemset in it also exists in the supported itemsets of size $k-1$.
  (b) If not every sub-itemset does, then prune it
(3) Now with the final candidates, determine if they have mininum support
(4) To determine association rules, find which itemsets have at least minimum confidence

6.7 Rules Possible

The number of ways of selecting i items from the k in a supported itemset of cardinality k for the right-hand side of a rule is given by:

$${}_iC_k$$

Total number of rules:

$${}_kC_{k-1}$$

$$2^k - 2$$

800 supported itemsets in C_2 if $800 \times \frac{799}{2}$.