Thesis Defense

Data Discretization Simplified:

Randomized Binary Search Trees for Data Preproccessing

Donald Joseph Boland Jr.

December 10th, 2007

Sound Bites

- **Classification** is a Useful Field in Data Mining.

Sound Bites

- **Classification** is a Useful Field in Data Mining.
- **Discretization** Helps to Minimize Classifier Confusion from Numeric Data and Increase Accuracy.

Sound Bites

- **Classification** is a Useful Field in Data Mining.
- **Discretization** Helps to Minimize Classifier Confusion from Numeric Data and Increase Accuracy.
- **DiscTree** is a New Discretization Algorithm Based on a **Randomized Binary Search Tree**.

Sound Bites

- **Classification** is a Useful Field in Data Mining.
- **Discretization** Helps to Minimize Classifier Confusion from Numeric Data and Increase Accuracy.
- **DiscTree** is a New Discretization Algorithm Based on a **Randomized Binary Search Tree**.
- This Thesis Implements DiscTree and Compares it to Other Frequently Used Discretization Methods.

Sound Bites

- **Classification** is a Useful Field in Data Mining.
- **Discretization** Helps to Minimize Classifier Confusion from Numeric Data and Increase Accuracy.
- **DiscTree** is a New Discretization Algorithm Based on a **Randomized Binary Search Tree**.
- This Thesis Implements DiscTree and Compares it to Other Frequently Used Discretization Methods.
- Results Lead to the Conclusion that There is No Single Best Method In All Cases.

Glossary

- **Instance** refers to one occurence in the data

Glossary

- **Instance** refers to one occurence in the data
- **Attribute** refers to one facet of an instance

Glossary

- **Instance** refers to one occurence in the data
- **Attribute** refers to one facet of an instance
- **Class** refers to the decision made for the instance

What is Classification?

- Start With a Set of Pre-Classified Example Instances

What is Classification?

- Start With a Set of Pre-Classified Example Instances
- Create a Theory/Concept for How Attributes Relate to Classes

What is Classification?

- Start With a Set of Pre-Classified Example Instances
- Create a Theory/Concept for How Attributes Relate to Classes
- Use/Test Theory on Future, Unforseen Instances

Useful Classification

- Student Data Used to Make Automated Financial Aid/Scholarship/Admissions Decisions

Useful Classification

- Student Data Used to Make Automated Financial Aid/Scholarship/Admissions Decisions
- Part Measurements Used to Make Accept/Reject Decision in Automated Manufacturing

Useful Classification

- Student Data Used to Make Automated Financial Aid/Scholarship/Admissions Decisions
- Part Measurements Used to Make Accept/Reject Decision in Automated Manufacturing
- Medical Test Data Used to Diagnose Specific Diseases/Conditions

Classification Methods

- Many Forms of Classification Methods Exist*, Including:
  - Decision Tree Learners (J48, C4.5)
  - Rule-Generating Learners (PRISM, RIPPER)
  - Instance-Based Learners (Nearest Neighbor, K-Means)

*For Further Explanation of Other Methods, see Thesis Document

Classification Methods

- Many Forms of Classification Methods Exist*, Including:
  - Decision Tree Learners (J48, C4.5)
  - Rule-Generating Learners (PRISM, RIPPER)
  - Instance-Based Learners (Nearest Neighbor, K-Means)
- However, for Controlled Experimental Purposes, Only One Classifer Used: Naïve Bayes Classifier

*For Further Explanation of Other Methods, see Thesis Document

The Basics

- Highly Studied Statistical Method of Classification

The Basics

- Highly Studied Statistical Method of Classification
- Originally a *Straw Man* Method

The Basics

- Highly Studied Statistical Method of Classification
- Originally a *Straw Man* Method
- Assumes Independence of Attributes

The Basics

- Highly Studied Statistical Method of Classification
- Originally a *Straw Man* Method
- Assumes Independence of Attributes
- Easily Handles Missing Attribute Values

The Basics

- Highly Studied Statistical Method of Classification
- Originally a *Straw Man* Method
- Assumes Independence of Attributes
- Easily Handles Missing Attribute Values
- Small Memory Footprint (only Keeps Value and Class Frequency Counts)

The Basics

- Highly Studied Statistical Method of Classification
- Originally a *Straw Man* Method
- Assumes Independence of Attributes
- Easily Handles Missing Attribute Values
- Small Memory Footprint (only Keeps Value and Class Frequency Counts)
- Makes Decisions Using Bayes' Theorem

Bayes' Theorem

- Simple View: $\mathit{next} = \mathit{old} \times \mathit{new}$

Bayes' Theorem

- Simple View: $next = old \times new$
- More Formally:

$$P(H|E) = \frac{P(H)}{P(E)} \prod_i P(E_i|H)$$

Where H is the class/hypothesis being considered and

E is the evidence of Current Conditions

Bayes' Theorem Explained

$$P(H|E) = \frac{P(H)}{P(E)} \prod_i P(E_i|H)$$

- Where $P(H)$ represents the prior probability of the class H;

Bayes' Theorem Explained

$$P(H|E) = \frac{P(H)}{P(E)} \prod_i P(E_i|H)$$

- Where $P(H)$ represents the prior probability of the class H;
- E_i represents the current evidence (attribute value);

Bayes' Theorem Explained

$$P(H|E) = \frac{P(H)}{P(E)} \prod_i P(E_i|H)$$

- Where $P(H)$ represents the prior probability of the class H;
- E_i represents the current evidence (attribute value);
- $P(E_i|H)$ represents the probability of attribute value E_I occuring with class H; and

Bayes' Theorem Explained

$$P(H|E) = \frac{P(H)}{P(E)} \prod_i P(E_i|H)$$

- Where $P(H)$ represents the prior probability of the class H;
- E_i represents the current evidence (attribute value);
- $P(E_i|H)$ represents the probability of attribute value E_I occuring with class H; and
- $P(H|E)$ represents the probability of class H given all the current evidence E, and is called the posterior probability.

Used for Classification

- For Each Instance, use its attribute values in the equation

Used for Classification

- For Each Instance, use its attribute values in the equation
- Each Class is Used in the Equation to Calculate its Posterior Probability

Used for Classification

- For Each Instance, use its attribute values in the equation
- Each Class is Used in the Equation to Calculate its Posterior Probability
- The Class with the Largest Posterior Probability is Selected as the Classification of the Instance

Why Choose Naïve Bayes

- Dougherty et. al Found that Each Form of Discretization Tried on Naïve Bayes classifiers Increased Performances

Why Choose Naïve Bayes

- Dougherty et. al Found that Each Form of Discretization Tried on Naïve Bayes classifiers Increased Performances
- Domingos and Pazzani Found Naïve Bayes classifiers with Discretization Out-Performed Other Methods and that the Attribute Independence Assumption did not Greatly Degrade Perfromance when used with Strongly Related Data

Why Choose Naïve Bayes

Many of the Most Recent Proposals for new Discretization have Been Proposed for Naïve Bayes classifiers ; Specifically, Webb puts Forth Many Methods Specifically for the Naïve Bayes classifiers . We Test Against one Called PKID.

What is Discretization

- Data comes in several forms
    - **Nominal** or qualitative data
    - **Ordinal** or nonquantitative ranked data
    - **Continuous**, numeric, or quantitative data

What is Discretization

- Data comes in several forms
  - **Nominal** or qualitative data
  - **Ordinal** or nonquantitative ranked data
  - **Continuous**, numeric, or quantitative data
- Quantitative Data can Cause Problems for Classifiers

What is Discretization

- Data comes in several forms
  - **Nominal** or qualitative data
  - **Ordinal** or nonquantitative ranked data
  - **Continuous**, numeric, or quantitative data
- Quantitative Data can Cause Problems for Classifiers
- Discretization Converts Numeric Data into Nominal Form

What is Discretization

- Data comes in several forms
  - **Nominal** or qualitative data
  - **Ordinal** or nonquantitative ranked data
  - **Continuous**, numeric, or quantitative data
- Quantitative Data can Cause Problems for Classifiers
- Discretization Converts Numeric Data into Nominal Form
- More Specifically, Discretization Replaces Numeric Values with Possibly Infite Values with a Fixed Set of Nominal Values.

How It Works

- Numeric Values are read, sorted, and placed in "buckets"
- Buckets or Bins Store to a fixed Range of Continuous Values.
- Data Values are Replaced by the Name of the Bucket They are Placed In

Methods

While Several Methods of Discretization are Reviewed*, We Experiment With Four:

- Equal Interval Width Discretization (EWD)

We Also Test Provide Results from Undiscretized data using the *cat* command

*For More Details on Discretization and Specific Methods, see Thesis Document

Methods

While Several Methods of Discretization are Reviewed*, We Experiment With Four:

- Equal Interval Width Discretization (EWD)
- Entropy-Minimization Discretization

We Also Test Provide Results from Undiscretized data using the *cat* command

*For More Details on Discretization and Specific Methods, see Thesis Document

Methods

While Several Methods of Discretization are Reviewed*, We Experiment With Four:

- Equal Interval Width Discretization (EWD)
- Entropy-Minimization Discretization
- Propotional K-Interval Discretization (PKID)

We Also Test Provide Results from Undiscretized data using the *cat* command

*For More Details on Discretization and Specific Methods, see Thesis Document

Methods

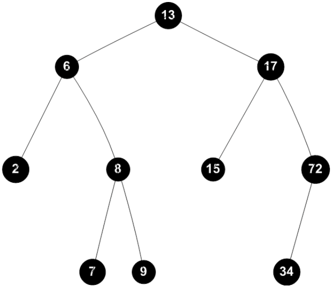
While Several Methods of Discretization are Reviewed*, We Experiment With Four:

- Equal Interval Width Discretization (EWD)
- Entropy-Minimization Discretization
- Propotional K-Interval Discretization (PKID)
- DiscTree

We Also Test Provide Results from Undiscretized data using the *cat* command

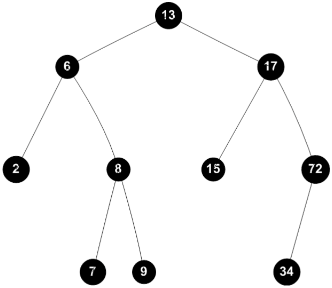
*For More Details on Discretization and Specific Methods, see Thesis Document

Randomized Binary Search Trees



- Like Binary Search Trees, But INSERT is Randomized

Randomized Binary Search Trees



- Like Binary Search Trees, But INSERT is Randomized
- At Root of Each Subtree, New Value Has $\frac{1}{T}$ Chance of Becoming Root, Where T is the Number of Instances at or Below Tested Node

DiscTree Premise

- Discretization Using Randomized Binary Search Trees as Base Data Structure

DiscTree Premise

- Discretization Using Randomized Binary Search Trees as Base Data Structure
- Tree Nodes Store a Value and Frequency Counts for Classes at and below Node

DiscTree Premise

- Discretization Using Randomized Binary Search Trees as Base Data Structure
- Tree Nodes Store a Value and Frequency Counts for Classes at and below Node
- Nodes with at Least $\sqrt{N}$, where N is the Number of Training Instances, at or below them can be substituted for continuous values.

Cross-Validation

- Cross-validation is a Statistical Method to Divide Data into a Fixed Number of Partitions, with Part for Training and Part for Testing
- Used to Generate Many Results of Classifier Runs, Rather than Relying on just One Run Each
- Performance is Averaged Across Several Runs, Preventing one Standout Result from Causing a Conclusion
- Experiment Utilized 10 by 10-fold Cross-validation, Generating 100 Results per Class per Discretization Method

Cross-Validation Explanation

Because We Used 24 Data Sets, We Generated Quite a Bit of Data. For a Data Set with Three Classes, For Example,

$$\begin{aligned} 5 \ \textit{Discretization Methods} \times 100 \ \textit{Results} \times 3 \ \textit{Classes} \\ = 300 \ \textit{Results per Discretization Method} \\ = 1500 \ \textit{Total Results} \end{aligned}$$

This Means that for the Letter Data Set, with 26 Classes, We Generated 2600 Results per Discretization Method, for a Total of 13000 Results.

Performance Measures

- **Accuracy**, or **acc**, Describes the Percentage of Cases Where The Learner/Method Pair makes the Correct Identification of a Instance's Class.

Performance Measures

- **Accuracy**, or **acc**, Describes the Percentage of Cases Where The Learner/Method Pair makes the Correct Identification of a Instance's Class.
- **Probability of Detection**, or **pd**, Describes the Percentage of the Target Class that is Correctly Identified.

Performance Measures

- **Accuracy**, or **acc**, Describes the Percentage of Cases Where The Learner/Method Pair makes the Correct Identification of a Instance's Class.
- **Probability of Detection**, or **pd**, Describes the Percentage of the Target Class that is Correctly Identified.
- **Probability of not False Alarm**, or **npf**, Describes The Percentage of the identified cases where an Identification of the Target Class is Correct

Performance Measures

- **Precision**, or **prec**, Describes the Proportion of Cases where Instances Identified as Being of a Particular Class actually Belong to that Class

Performance Measures

- **Precision**, or **prec**, Describes the Proportion of Cases where Instances Identified as Being of a Particular Class actually Belong to that Class
- **Balance**, or **bal**, Describes the balance of Probability of Detection and Probability of False Alarm. A Higher Balance means the Learner is Identifying Most Instances Correctly Without Risking False Alarms to be Correct.

Mann-Whitney U-test

- Non-parametric Measure to Compare Learner/Method Pair
- Makes No Assumptions about Shape of Data
- Allows Comparison of Results With Differing Number of Values
- Requires no Post-Processing to Explain Results

DiscTree Comparison Results

- Two Features of DiscTree were Questioned During Implementation

DiscTree Comparison Results

- Two Features of DiscTree were Questioned During Implementation
  - Nominal Value Discretization

DiscTree Comparison Results

- Two Features of DiscTree were Questioned During Implementation
  - Nominal Value Discretization
  - Garbage Collection

DiscTree Comparison Results

- Two Features of DiscTree were Questioned During Implementation
  - Nominal Value Discretization
  - Garbage Collection
- To Determine Best Method, Coded Each and Compared Using Described Experimental Design

DiscTree Comparison Results

- Methods Performed Vary Similarly; However,

DiscTree Comparison Results

- Methods Performed Vary Similarly; However,
  - **disctree3**, the method using just Garbage Collection, performed most accurately

DiscTree Comparison Results

- Methods Performed Vary Similarly; However,
  - **disctree3**, the method using just Garbage Collection, performed most accurately
  - **disctree3** and **disctree4** (which implemented neither Garbage Collection nor Nominal Discretization) beat **disctree2** which implemented both.

DiscTree Comparison Results

- Methods Performed Vary Similarly; However,
  - **disctree3**, the method using just Garbage Collection, performed most accurately
  - **disctree3** and **disctree4** (which implemented neither Garbage Collection nor Nominal Discretization) beat **disctree2** which implemented both.
- Because it Acquired the Most U-test Wins, **disctree3** was Selected for use in the General Comparsion.

Method Comparison Results

	acc	bal	npf	pd	prec
cat	0	0	0	0	0
disctree3	1	2	0	2	1
fayyadIrani	4	4	4	4	4
pkid	1	2	0	2	1
tbin	1	1	3	0	1

Figure: Summary of *U*-test Results

Method Comparison Results

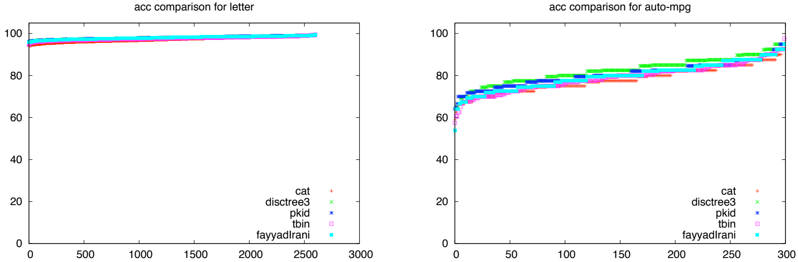
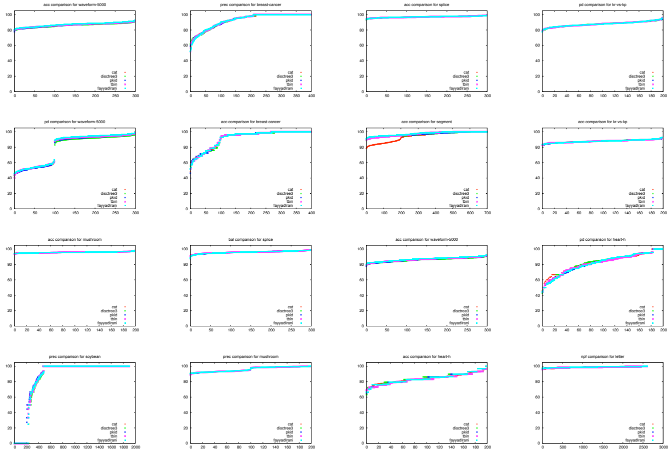


Figure: Sample Normal(left) and Standout(right) Results

Method Comparison Results



Possible Future Work Areas

- Convert DiscTree Algorithm to an Incremental Discretization Method

Possible Future Work Areas

- Convert DiscTree Algorithm to an Incremental Discretization Method
- Implementing DiscTree Algorithm on Other Tree Data Structures

Possible Future Work Areas

- Convert DiscTree Algorithm to an Incremental Discretization Method
- Implementing DiscTree Algorithm on Other Tree Data Structures
- Addition of Additional Data Preprocessing

Possible Future Work Areas

- Convert DiscTree Algorithm to an Incremental Discretization Method
- Implementing DiscTree Algorithm on Other Tree Data Structures
- Addition of Additional Data Preprocessing
- Reexamination of DiscTree Algorithm for "Best Values"

Conclusions From This Thesis

- Across All Performance Measures, Entropy-Minimization out performs the competition according to U-test Results.

Conclusions From This Thesis

- Across All Performance Measures, Entropy-Minimization out performs the competition according to U-test Results.
- However, in Most Cases, Other Methods Perform Very Nearly as well as the Entropy-Minimization Method.

Conclusions From This Thesis

- Across All Performance Measures, Entropy-Minimization out performs the competition according to U-test Results.
- However, in Most Cases, Other Methods Perform Very Nearly as well as the Entropy-Minimization Method.
- DiscTree Performs Second-Best in Each Performance Measure

Conclusions from This Thesis

Results Lead Us to Believe:

- There is No Single Best Method of Discretization in All Cases;

Conclusions from This Thesis

Results Lead Us to Believe:

- There is No Single Best Method of Discretization in All Cases;
- However, Discretization Almost Always Increases Accuracy and Other Performance Measures in Naïve Bayes classifiers, with simple methods performing nearly as well; and,

Conclusions from This Thesis

Results Lead Us to Believe:

- There is No Single Best Method of Discretization in All Cases;
- However, Discretization Almost Always Increases Accuracy and Other Performance Measures in Naïve Bayes classifiers, with simple methods performing nearly as well; and,
- Perhaps the Energy Spent Continuing to Study Batch Discretization Might Be Better Spent Elsewhere.

Questions?