

FAST KERNEL-DENSITY-BASED CLASSIFICATION  
AND CLUSTERING USING P-TREES

A Thesis  
Submitted to the Graduate Faculty  
of the  
North Dakota State University  
of Agriculture and Applied Science

By

Anne Margarete Denton

In Partial Fulfillment of the Requirements  
for the Degree of  
MASTER OF SCIENCE

Major Department:  
Computer Science

March 2003

Fargo, North Dakota

## ABSTRACT

Denton, Anne Margarete, M.S., Department of Computer Science, College of Science and Mathematics, North Dakota State University, March 2003. Fast Kernel-density-based Classification and Clustering Using P-trees. Major Professor: Dr. William Perrizo.

Much hardware development effort is dedicated to fast bit-wise operations on large amounts of data, as is necessary for image and video processing. Database and data mining applications do not normally use this processing power. Peano Count Tree-based algorithms are an exception in replacing database scans by AND operations on a compressed bit-vector representation of the data. In this thesis, we show how AND operations on Peano Count Trees, or P-trees, can be implemented; describe an application programmer interface to use them; and develop a variety of data mining algorithms that are based on them. Two types of classification algorithms as well as one clustering algorithm that use ideas from traditional algorithms, adapt them to the P-tree setting, and introduce new improvements are described. All algorithms are fundamentally based on kernel-density estimates that can be seen as a unifying concept for much of the work done in classification and clustering. The two classification algorithms in this thesis differ in their approach to handling data with many attributes. Paper 1 demonstrates means of selecting the most important attributes, thereby reducing the space in which densities have to be evaluated. Paper 2 solves the problem of high dimensionality by using an assumption on independence of attributes. Highly correlated attributes are identified and joined in a novel way. Paper 3 describes a clustering algorithm that combines ideas from three traditional clustering techniques into one P-tree-based method. For all algorithms, we show where they outperform traditional methods.

## **ACKNOWLEDGMENTS**

I would like to thank my adviser, Dr. William Perrizo, for his encouragement and support, and for creating an excellent research environment. Thanks to the other committee members, Dr. D. Bruce Erickson, Dr. Paul Juell, and Dr. Dogan Comez, for their interest in my thesis and for valuable comments they made. Thanks to all DataSURG members for their willingness to explore research ideas together. Last but not least, thanks to Alan, Carl, and Martha whose understanding and support made this thesis possible.

# TABLE OF CONTENTS

ABSTRACT .....	iii
ACKNOWLEDGMENTS .....	iv
LIST OF TABLES .....	ix
LIST OF FIGURES .....	x
CHAPTER 1. INTRODUCTION .....	1
1.1. Organization of the Thesis .....	2
1.2. References .....	4
CHAPTER 2. DATA MINING USING P-TREE RELATIONAL SYSTEMS .....	6
2.1. Rows versus Columns .....	6
2.2. Columns of Bits .....	8
2.3. Distance Measures and Columns of Bits .....	10
2.4. Concept Hierarchies and Concept Slices .....	12
2.5. Partitions .....	14
2.6. Column-based Structures and Software Engineering .....	15
2.7. References .....	17
CHAPTER 3. P-TREES: CONCEPTS, IMPLEMENTATION, AND APPLICATION PROGRAMMING INTERFACE .....	19
3.1. Concepts .....	19
3.2. Implementation .....	29
3.3. Application Programming Interface (API) .....	36
3.4. P-tree API as an Example of a Column-based Design .....	42
3.5. References .....	44

CHAPTER 4. LAZY KERNEL-DENSITY-BASED CLASSIFICATION USING P-TREES .....	45
4.1. Introduction .....	45
4.2. Kernel Methods .....	46
4.3. Classification Strategies .....	50
4.4. Experimental Results .....	56
4.5. Remarks to Simplify the Transition Between Chapters 4, 5, and 6 .....	57
4.6. References .....	58
CHAPTER 5. PAPER 1: FAST RULE-BASED CLASSIFICATION USING P-TREES .....	61
5.1. Abstract .....	61
5.2. Introduction .....	61
5.3. Algorithm .....	63
5.4. Implementation and Results .....	73
5.5. Conclusions .....	79
5.6. References .....	80
CHAPTER 6. PAPER 2: A KERNEL-BASED SEMI-NAIVE BAYES CLASSIFIER USING P-TREES .....	83
6.1. Abstract .....	83
6.2. Introduction .....	83
6.3. Naive and Semi-naive Bayes Classifier Using Kernel Density Estimation ....	85
6.4. Implementation and Results .....	94
6.5. Conclusions .....	102
6.6. References .....	102

CHAPTER 7. PAPER 3: EFFICIENT HIERARCHICAL CLUSTERING OF LARGE DATA SETS USING P-TREES .....	105
7.1. Abstract .....	105
7.2. Introduction .....	105
7.3. Taking a Fresh Look at Established Algorithms .....	107
7.4. Hierarchical Algorithm .....	111
7.5. Algorithm .....	113
7.6. Performance Analysis .....	116
7.7. Conclusions .....	118
7.8. References .....	119
CHAPTER 8. CONCLUSIONS AND OUTLOOK .....	121
8.1. Conclusions .....	121
8.2. Outlook .....	123
8.3. References .....	124
APPENDIX A. FORMAL DEFINITIONS OF P-TREE REPRESENTATIONS .....	125
A.1. P-tree Definition .....	125
A.2. AND Operation .....	128
A.3. Pre-order Sequence Representation .....	129
A.4. Array-sequence Equivalence .....	130
A.5. Bit-vector Representation .....	132
A.6. Dense P-trees .....	134
A.7. Array-converted Dense P-trees .....	136
A.8. Comparison with Implemented Code .....	137

A.9. References .....	138
APPENDIX B. SIGNIFICANCE AND THE EXACT FORMULATION OF INFORMATION GAIN .....	139
B.1. Significance .....	140
B.2. Information Gain .....	140

## LIST OF TABLES

<u>Table</u>	<u>Page</u>
3.1. Bit order in generalized Peano-order sorting .....	25
3.2. Example of a data set that was sorted using the bit-order in Table 1 .....	26
5.1. Properties of data sets .....	75
5.2. Results of rule-based classification .....	76
6.1. Summary of data set properties .....	96
6.2. Results of different Naive and Semi-naive Bayes algorithms .....	97
7.1. Comparison of cluster centers for the data set of Figure 7.3 .....	118



## LIST OF FIGURES

<u>Figure</u>	<u>Page</u>
3.1. Space-filling curves .....	20
3.2. Construction of a Peano-ordered sequence through interleaving of bits.....	21
3.3. Number of P-tree nodes for different sorting schemes (data sets as explained in Chapter 4, with the crop data set restricted to $3 \cdot 10^5$ data points) .....	26
3.4. Time for the classification of 100 data points using rule-based classification as explained in Chapter 4 .....	27
3.5. Structure of a P-tree .....	29
3.6. Representations of P-tree structure (pure 1 information displayed) .....	33
3.7. Large example that represents the implementation for this thesis .....	36
3.8. Relationships among the most important classes in the P-tree API .....	37
4.1. Error rate comparison among the algorithms of Chapters 4-6 .....	56
5.1. Peano order sorting and P-tree construction .....	65
5.2. <i>InfoGain</i> and <i>ExactInfoGain</i> for $x = 250$ , $y = 50$ , and $x_1 = 50$ .....	70
5.3. Difference between <i>ExactInfoGain</i> and <i>InfoGain</i> . .....	71
5.4. Difference between a vote based on 20 paths and a vote based on 1 path in the rule-based algorithm in units of the standard error .....	77
5.5. Difference between standard and exact information in the rule-based algorithm in units of the standard error .....	78
5.6. Scaling of execution time as a function of training set size .....	79
6.1. Comparison between Gaussian distribution function and kernel density estimate.....	88
6.2. Peano order sorting and P-tree construction .....	92
6.3. Decrease in error rate for the P-tree Naive Bayes classifier compared with the traditional Naive Bayes classifier in units of the standard error .....	98

6.4.	Decrease in error rate for 3 parameter combinations for the Semi-naive Bayes classifier compared with the P-tree Naive classifier .....	99
6.5.	Scaling of execution time as a function of training set size .....	101
7.1.	Energy landscape (black) and potential of individual data items (gray) for a Gaussian influence function .....	110
7.2.	8-by-8 image and its P-tree .....	114
7.3.	Speed comparison between our approach and k-means .....	117
7.4.	Histogram of values used to compare our approach with k-means .....	117