Supervised and unsupervised discretization of Continuous Features

Ron Kohavi

Joint work with James Dougherty and Mehran Sahami

Talk Outline

☞ ① Introduction, definitions, motivation.

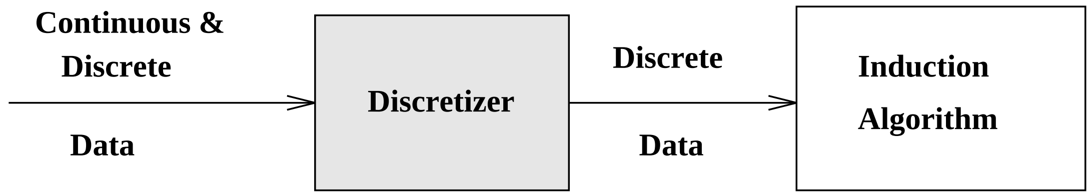
② Outline of the three discretization methods compared.

③ Experimental Results.

④ Related work, future work & Summary.

Introduction

1. A discretization algorithm converts continuous features into discrete features.



2. No clever processing of data can improve the information that X contains about Y [Data processing inequality theorem].
3. A good discretization algorithm should not increase the Bayes-error rate much.

Motivation

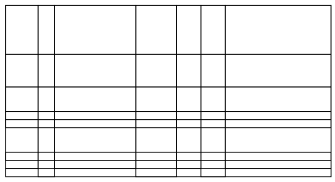
Why study discretization?

1. Some algorithms are limited to discrete inputs.
2. Many algorithms discretize as part of the learning algorithm (*e.g.*, decision trees). Could this part be improved?
3. Efficiency. Continuous features drastically slows decision tree induction [Catlett].
4. Ability to view the data (General Logic Diagrams).

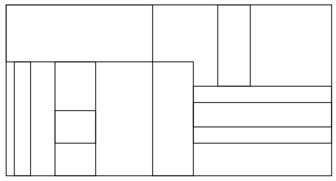
Classifying Discretization Algorithms

Discretization can be classified on two dimensions:

1. Supervised vs. unsupervised.
   Supervised discretization uses class information.
2. Global (mesh) vs. local



Global



Local

One can apply some discretization methods either globally or locally.

Single Feature Discretization

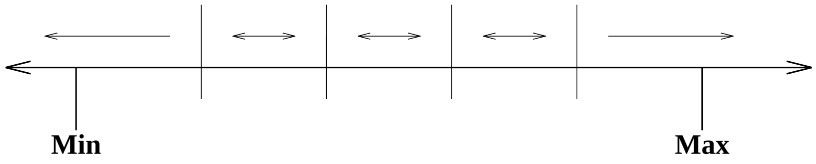
In this talk, we consider discretizing each feature separately (global discretization) for the following reasons:

1. It limits the scope of the problem.
   Allowing discretization of multiple input features is as hard as the original induction algorithm (map the features to the discrete number corresponding to the correct class).

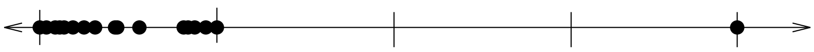
2. Easy to interpret discretization (comprehensibility).

Equal Interval Width (Binning)

1. Given the number of bins k, divide the training-set range into k equal-sized bins.

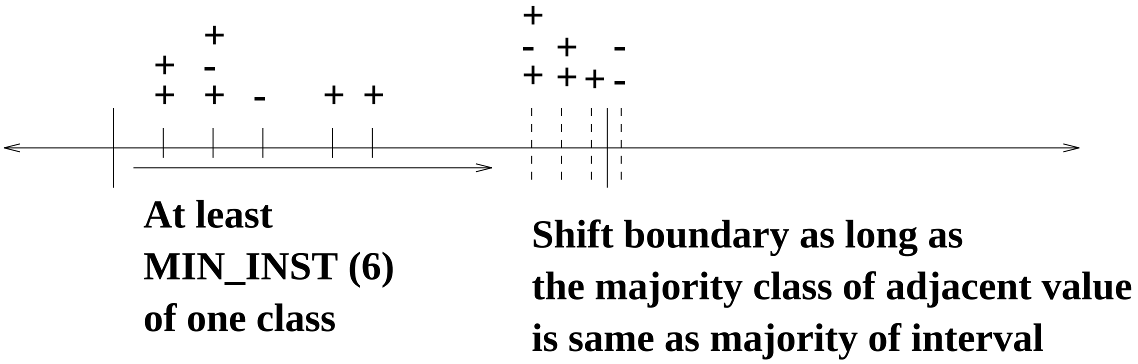


2. Problems:
   (a) Unsupervised.
   (b) Where does k come from?
   (c) Sensitive to outliers



Holte's OneR

Holte (1993) described a discretization method used in the OneR induction algorithm, which induced one-level decision trees (decision stumps). The method works as follows:



This method is supervised.

Minimal Entropy Partitioning

A method similar to building a decision tree based on a single feature.

1. Find the best threshold split, such that the mutual-information between the feature and the label is maximal.
2. Split the data into two according to threshold.
3. Recursively discretize each partition.

Question: how many partitions?
Suggestions: D2 [Catlett], MDL approach [Fayyad & Irani].

The method is supervised.

Experimental Setup

1. We compared equal width interval binning ($k = 10$ intervals and $k = 2 \cdot \log \ell$), Holte's 1R (MIN_INST=6), and the entropy-based partitioning (Fayyad & Irani).
2. We compared C4.5 and Naive-Bayes with and without the filter discretizations.
3. We chose 16 datasets from U.C. Irvine, all containing at least one continuous feature.

   For datasets with more than $3,000$ instances, we split the data to train/test (2/3, 1/3) and for the rest we ran 5-fold cross-validation.

The Wrong Way to Experiment with Discretization

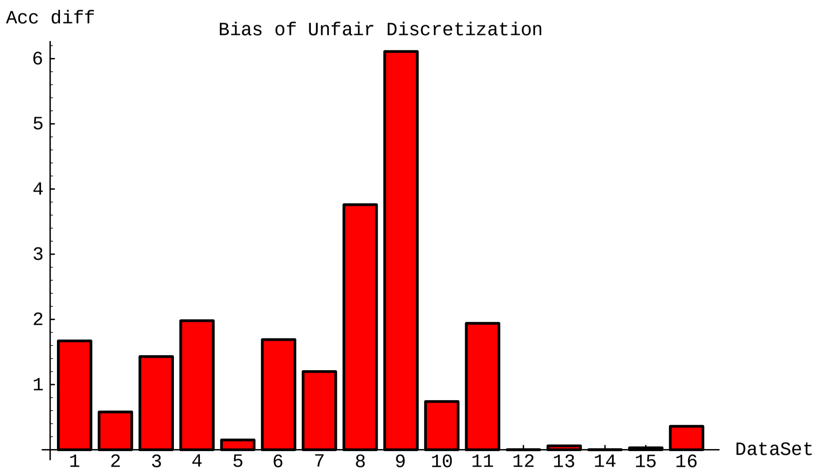
What's wrong with the following experimental methodology:

1. Discretize the data file.
2. Run ten-fold cross-validation (or 1/3, 2/3 holdout).
3. Report cross-validation accuracy with discretization and without.

?

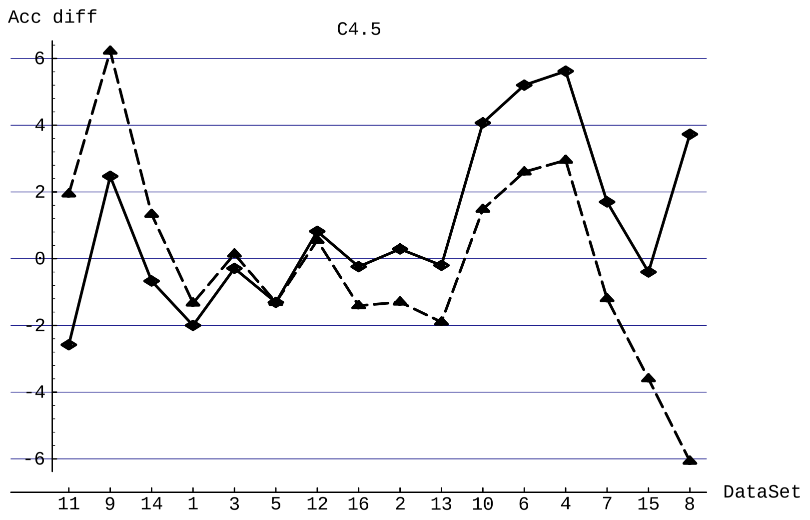
The Wrong Way to Experiment with Discretization

Problem: The discretization process must only use the training set. The test instances in each fold of CV must be hidden.



Results : C4.5

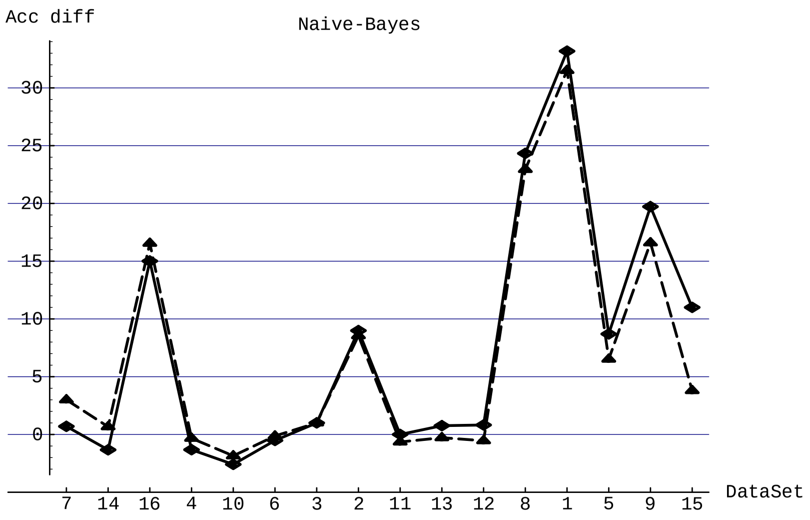
The graph shows accuracy difference of pre-discretized data minus original C4.5.



Solid line: Entropy. Dashed line: $\log \ell$-binning.

Results : Naive-Bayes

The graph shows accuracy difference of pre-discretized data minus original Naive-Bayes.



Solid line: Entropy. Dashed line: $\log \ell$-binning.

Related Work: I

1. *Equal frequency intervals* : Assign approximately equal number of instances to each bin.
2. *Maximal marginal entropy* : Use labels to shift boundaries in order to decrease the entropy of intervals [Chmielewski & GrzymalaBusse 1994, Wong & Chiu 1987].
3. ChiMerge by Kerber (1992): bottom-up approach. Intervals are combined based on χ^2 test.
   StatDisc by Richeldi & Rossotto (1995) extends ChiMerge by merging N adjacent intervals.
4. Pfahringer (1995) : MDL approach & best-first search.

Related Work: II

1. Maass (1994); Fulton, Kasif & Salzberg (1995): use dynamic programming to find optimal thresholds that minimize errors.
2. Cluster-based approach by Chmielewski and Grzymala-Busse (1994).
3. Adaptive quantizers by Chan, Batur & Srinivasan (1991).
4. Monothetic Contrast Criterions by Van de Merckt (1993).
5. Predicative value maximization by Weiss, Galen & Tadepalli (1990).
6. Vector quantization by Kohonen (1989).

Future Work

1. When is global discretization better than local disc? Global discretization reduces variance.
2. Large differences in number of discretization intervals between methods. For example, on the diabetes dataset

Method	Accuracy	Intervals per attr							
Entropy :	76.04	2	4	1	2	3	2	2	3
Holte's 1R:	72.40	6	13	4	6	18	25	41	12
$\log \ell$ binning:	73.44	8	14	11	11	15	16	18	11

3. Attempt to optimize number of intervals using a wrapper approach.
4. Use grow/prune approach to determine number of intervals as in decision trees.
5. Compare with other methods.

Summary

1. We reviewed discretization methods and compared three of them, one unsupervised and two supervised.
2. We used the filter-model that generates global discretization, but methods could be used internally for local discretization.
3. Performance for C4.5 slightly improved using entropy-discretization, probably due to reduced variance in estimating the thresholds using all the data for each feature.
4. All methods significantly helped Naive-Bayes, but entropy was also the best here.
5. The discretization code is publicly available as part of $\mathcal{MLC}$++, the Machine Learning library in C++ developed at Stanford.