Discretization of Ordinal Attributes and Feature Selection

Huan Liu and Rudy Setiono

Department of Information Systems and Computer Science

National University of Singapore

Kent Ridge, Singapore 0511

{liuh,rudys}@iscs.nus.sg

Abstract

The performance of classification algorithms may deteriorate due to irrelevant attributes. Numeric attributes make the situation worse, since many classification algorithms require that the training data contain only discrete attributes. Discretization can turn numeric attributes into discrete ones. Feature selection can eliminate some irrelevant attributes. This paper describes Chi2, a simple and general algorithm that uses the χ^2 statistic to discretize numeric attributes repeatedly until some inconsistencies are found in the data, and achieves feature selection via discretization. In addition, it can handle mixed attributes and multi-class data naturally. Experiments are conducted on the real and synthetic data sets. The empirical results demonstrate that Chi2 is effective in feature selection and discretization of numeric and ordinal attributes.

1 Introduction

Feature selection is a task to select the minimum number of attributes needed to represent the data accurately. By using relevant features, classification algorithms can in general improve their predictive accuracy, shorten the learning period, and result in the simpler concepts. There are abundant feature selection algorithms. Some use statistical methods such as principle component to compose a smaller number of new features (a summary in [13]); some select a subset of the original attributes (a summary in [7]). This paper considers the latter since it not only has the above virtues, but also serves as an indicator on what kind of data (along those selected features) should be collected. In the latter category of feature selection, the algorithms can be further divided based on the data types they operate on. The basic two types of data are nominal (e.g., attribute *color* may have values of red, green, yellow) and ordinal (e.g., attribute *winning position* can have values of 1, 2, and 3, or attribute *salary* can have 22345.00, 46543.89, etc. as its values). Many feature selection algorithms [1, 12, 3, 7] are shown to work effectively on discrete data or even more strictly, on binary data (and/or binary class value). In order to deal with numeric attributes, a common practice for those algorithms is to discretize the data before conducting feature selection. This paper provides a way to select features directly from numeric attributes while discretizing them. Numeric data are very common in real world problems. However, many classification algorithms require that the training data contain only discrete attributes, and some would work better on discretized or binarized data [2, 6]. If those numeric data can be *automatically* transformed into discrete ones, these classification algorithms would be readily at our disposal. Chi2 is our effort towards this goal: discretize the numeric attributes as well as select features among them.

The problem this work tackles is as follows: there are data sets with numeric attributes, some are irrelevant and the range of each numeric attribute could be very wide; find an algorithm that can automatically discretize the numeric attributes as well as remove those irrelevant ones.

This work stems from Kerber's ChiMerge [6] which is designed to discretize numeric attributes based on the χ^2 statistic. ChiMerge consists of an initialization step and a bottom-up merging process, where intervals are continuously merged until a termination condition, which is determined by a significance level α (set manually), is met. It is an improvement from the most obvious simple methods such as *equal-width-intervals*, which divides the number line between the minimum and maximum values into N intervals of equal size; or *equal-frequency-intervals*, in which the interval boundaries are chosen so that each interval contains approximately the same number of training examples. Instead of defining a width or frequency threshold (which is not easy until scrutinizing each attribute and knowing what it is), ChiMerge requires α to be specified. Nevertheless, too big or too small an α will over- or under-discretize an attribute. An extreme example of under-discretization is the continuous attribute itself. Over-discretization will introduce many inconsistencies[1] nonexistent before, thus change the characteristics of the data. In short, it is not easy to find a proper α for ChiMerge. It is thereby ideal to let the data determine what value α should take. This leads to Phase 1 of our Chi2 algorithm. Naturally, if the discretization continues without generating more inconsistencies than in the original data, it is possible that some attributes will be discretized into one interval only. Hence, these attributes can be removed without affecting the discriminating power of the original data.

In the following section, we describe the Chi2 algorithm in sufficient detail so that a reader can reimplement it easily. A detailed run for the Iris data is given in Section 3 to explain the Chi2 behavior; the ultimate verification is performed through/against a well-known classifier C4.5, experimenting on three data sets with various types of attributes. In Section 4, we discuss issues such as related work, computational complexity, Chi2's limitations, and its possible uses. The final section concludes our work.

[1] By inconsistency we mean that two patterns are the same, but classified into different categories.

2 Chi2 Algorithm

The Chi2 algorithm (summarized on the next page) is based on the χ^2 statistic, and consists of two phases. In the first phase, it begins with a high significance level (sigLevel), e.g., 0.5, for all numeric attributes of discretization. Each attribute is sorted according to its values. Then the following is performed: 1. calculate the χ^2 value as in equation (1) for every pair of adjacent intervals (at the beginning, each pattern is put into its own interval that contains only one value of an attribute); 2. merge the pair of adjacent intervals with the lowest χ^2 value. Merging continues until all pairs of intervals have χ^2 values exceeding the parameter determined by sigLevel (initially, 0.5, its corresponding χ^2 value is 0.455 if the degree of freedom is 1, more below). The above process is repeated with a decreased sigLevel until an inconsistency rate, δ is exceeded in the discretized data. Phase 1 is, as a matter of fact, a generalized version of ChiMerge of Kerber [6]. Instead of specifying a χ^2 threshold, Chi2 wraps up ChiMerge with a loop that automatically increments the χ^2 threshold (decrementing sigLevel). A consistency checking is also introduced as a stopping criterion in order to guarantee that the discretized data set accurately represents the original one. With these two new features, Chi2 automatically determines a proper χ^2 threshold that keeps the fidelity of the original data.

Phase 2 is a finer process of Phase 1. Starting with sigLevel0 determined in Phase 1, each attribute i is associated with a sigLevel[i], and takes turns for merging. Consistency checking is conducted after each attribute's merging. If the inconsistency rate is not exceeded, sigLevel[i] is decremented for attribute i's next round of merging; otherwise attribute i will not be involved in further merging. This process is continued until no attribute's values can be merged. At the end of Phase 2, if an attribute is merged to only one value, it simply means that this attribute is not relevant in representing the original data set. As a result, when discretization ends, feature selection is accomplished.

Chi2 Algorithm:

Phase 1:

```
set sigLevel = .5;
do while (InConsistency(data) < δ) {
   for each numeric attribute {
       Sort(attribute, data);
       chi-sq-initialization(attribute, data);
       do {
          chi-sq-calculation(attribute,data)
       } while (Merge(data))
   }
   sigLevel0 = sigLevel;
   sigLevel = decreSigLevel(sigLevel);
}
```

Phase 2:

```
set all sigLvl[i] = sigLevel0 for attribute i;
do until no-attribute-can-be-merged {
   for each mergeble attribute i {
      Sort(attribute, data);
      chi-sq-initialization(attribute, data);
      do {
         chi-sq-calculation(attribute,data)
      } while (Merge(data))
      if (InConsistency(data) < δ)
         sigLvl[i] = decreSigLevel(sigLvl[i]);
      else
         attribute i is not mergeble;
   }
}
```

The formula for computing the χ^2 value is:

$$\chi^2 = \sum_{i=1}^{2} \sum_{j=1}^{k} \frac{(A_{ij} - E_{ij})^2}{E_{ij}} \tag{1}$$

where:

k = number of classes,

A_{ij} = number of patterns in the ith interval, jth class,

R_i = number of patterns in the ith interval = $\sum_{j=1}^{k} A_{ij}$,

C_j = number of patterns in the jth class = $\sum_{i=1}^{2} A_{ij}$,

N = total number of patterns = $\sum_{i=1}^{2} R_i$,

E_{ij} = expected frequency of $A_{ij} = R_i * C_j / N$.

If either R_i or C_j is 0, E_{ij} is set to 0.1. The degree of freedom of the χ^2 statistic is one less the number of classes.

An efficiency issue is the Chi2's two-phase implementation. Phase 1 is simply designed for efficiency, specially when m is large. Due to the consistency checking which takes $O(n^2)$, the saving can be as much as $(m-1)n^2$ for each outermost loop by implementing the two phases.

3 Experiments

This section first briefly explains the data sets to be used, then shows in steps how Chi2 works on one of the data sets – the Iris data. Whether Chi2 is effective in feature selection and discretization is demonstrated experimentally.

Two sets of experiments are conducted. In the first set of experiments, we want to establish that 1. Chi2 helps improve predictive accuracy; and 2. Chi2 properly and effectively discretizes data as well as eliminates some irrelevant attributes because of 1. C4.5 [10] (an extension of ID3 [9]) is used to verify the effectiveness of Chi2. The reasons for our choice are 1. C4.5 (or ID3) works well for many problems and is well known, thus requiring no further description; and 2. C4.5 selects relevant features by itself in tree branching so it can be used as a benchmark, as in [7, 11, 1], to verify the effects of Chi2.

In the second set of experiments, we have a closer examination of Chi2's ability of discretization and feature selection by introducing a synthetic data set and adding noise attributes to the existing data set. Through these more controlled data sets, we can better understand how effective Chi2 is.

3.1 Data sets

3.1.1 Real data

Three data sets used in experiments are Iris, Wisconsin Breast Cancer and Heart Disease[2]. They have different types of attributes. The Iris data are of continuous attributes, the breast cancer data are of ordinal discrete ones, and the heart disease data have mixed attributes (numeric and discrete). The three data sets are described briefly as follows.

- **Iris data** contains 50 patterns each of the classes *Iris setosa*, *Iris versicolor*, and *Iris virginica*. Each pattern is described using four numeric attributes: *sepal-length*, *sepal-width*, *petal-length*, and *petal-width*. The originally odd numbered data are selected for training, the rest for testing.

- **Breast cancer data** contains 699 samples of breast fine-needle aspirates collected at the University of Wisconsin Hospital. There are 9 discrete attributes valued on a scale of 1 to 10. The class value is either benign or malignant. The data set is split randomly into two sets, 350 patterns for training and 349 for testing.

- **Heart disease data** contains medical cases of heart diseases. It contains numerically-valued features; there are 8 nominally-valued and 5 numerically-valued attributes. Two class values are: healthy and diseased heart. Removing patterns with missing attribute values, we use

[2]They are all obtained from the University of California-Irvine, machine learning repository via anonymous ftp to *ics.uci.edu*.

299 patterns in which one third are randomly chosen for testing, and the rest are for training.

3.1.2 Controlled data

Two extra data sets are designed to test if noise attributes can be removed. One is synthetic, the other is the Iris data added with noise attributes.

The synthetic data consists of 600 items and is described by four attributes among which only one attribute determines each item's class label. The values, v_0 of attribute A_0 are generated from a uniform distribution between the lower bound (L = 0) and the upper bound (U = 75), each item's class label is determined as follows: $v_0 < 25 \rightarrow$ class 1, $v_0 < 50 \rightarrow$ class 2, $v_0 < 75 \rightarrow$ class 3. Then we add noise attributes A_1, A_2, and A_3. The values of A_1 are generated from a normal distribution with $\mu = U/2$ (i.e. 37.5) and $\sigma = \mu/3$. The values of A_2 are generated from two normal distributions with $\mu = U/3$ (i.e. 25), $\mu = 2 * U/3$ (i.e. 50) and $\sigma = \mu/3$ respectively, 300 items each distribution. The values of A_3 are generated from a uniform distribution.

The second data is a modified version of Iris data. Four noise attributes A_4, A_5, A_6 and A_7 are added to the Iris training data corresponding to the four original attributes. The values of each noise attribute are determined by a normal distribution with $\mu = ave$ and $\sigma = (max - min)/6$, where ave is the average value of, max and min are the maximum and minimum values of the original attribute. The choice of $sigma$ is to approximate $\mu/3$ if the corresponding original attribute is of uniform distribution. Now there are total eight attributes, still 75 items.

3.2 Example

In this section, some steps of Chi2 processing for the Iris data are shown to demonstrate the behavior of Chi2. Table 1 shows the intervals, class frequencies, and χ^2 values of sepal-length after the initialization in Phase 1. The results for sepal-length after Phase 1 and Phase 2 are shown in Table 2. The

Int	Class Freq			χ^2	Int	Class Freq			χ^2
4.4	3	0	0	0.20	6.0	0	2	1	1.43
4.6	2	0	0	0.20	6.1	0	0	1	0.54
4.7	1	0	0	0.20	6.2	0	1	2	0.14
4.8	3	0	0	1.97	6.3	0	2	3	0.14
4.9	1	0	1	2.62	6.4	0	1	2	0.16
5.0	3	1	0	0.10	6.5	0	1	3	1.97
5.1	3	1	0	0.70	6.6	0	1	0	2.50
5.2	2	0	0	0.20	6.7	0	1	4	0.73
5.3	1	0	0	0.41	6.8	0	1	1	0.10
5.4	3	1	0	1.32	6.9	0	1	1	0.85
5.5	1	2	0	1.66	7.0	0	1	0	2.10
5.6	0	4	0	2.50	7.1	0	0	1	0.20
5.7	1	1	0	1.28	7.4	0	0	1	0.20
5.8	1	2	2	1.20	7.7	0	0	2	
5.9	0	1	0	0.54					

Table 1: The initial intervals, class frequencies, and χ^2 values for sepal-length.

inconsistency rate $\delta = 5\%$ is allowed in the experiment, that means up to 3 (75x0.05) inconsistencies are acceptable. The results for the other three attributes can be found in Appendix. Phase 1 stops at sigLevel = 0.2, $\chi^2 =$ 3.22. That means the next sigLevel (0.1) will introduce more inconsistencies. When Phase 2 terminates, the values of both sepal-length and sepal-width are merged into one value, so they can be removed; and attributes petal-length and petal-width are discretized into four discrete values each. With the χ^2 threshold 3.22, for example, six discrete values are needed for attribute sepal-length: $< 4.4 \rightarrow 0$, $< 4.9 \rightarrow 1$, ..., $< 6.1 \rightarrow 4$, and $\geq 6.1 \rightarrow 5$. The last one reads if a numeric value is greater than and equal to 6.1, it is quantized to 5.

Int	Class Freq			χ^2
4.4	9	0	0	5.05
4.9	1	0	1	8.11
5.0	12	3	0	13.64
5.5	3	12	3	14.23
6.1	0	10	21	

(a)

Int	Class Freq			χ^2
4.4	25	25	25	

(b)

Table 2: The intervals, class frequencies, and χ^2 values for attribute sepal-length after Phase 1 and Phase 2. The χ^2 thresholds are (a) 3.22 and (b) 50.6 respectively.

3.3 Empirical results on real data

First we show that after discretization, the number of attributes decreases for the three data sets (in Figure 1). For the Iris data, the number of attributes is reduced from 4 to 2 (petal length and petal width), each has four values (refer to Appendix). For the breast cancer data, 3 attributes are removed from the original 9 attributes. The remaining 6 attributes have 3, 4, 4, 5, 3, and 3 discrete values respectively. For the heart disease data, the discrete attributes are left out in discretization and feature selection although they are used for consistency checking. Among the 5 continuous attributes (1, 4, 5, 8 and 10), only 2 attributes (5 and 8) should remain as suggested by Chi2, having 8 and 4 discrete values respectively. For the cancer and disease data sets, the default inconsistency rate is used, i.e., 0.

Second, we run C4.5 on both the original data sets and the dimensionally reduced ones. C4.5 is run using its default setting. Chi2 discretizes the training data and generates a mapping table, based on which the testing data are discretized.

Shown in Figure 2 are predictive accuracies and tree sizes of C4.5 for the three data sets. Predictive accuracy improves and tree size drops (by half)

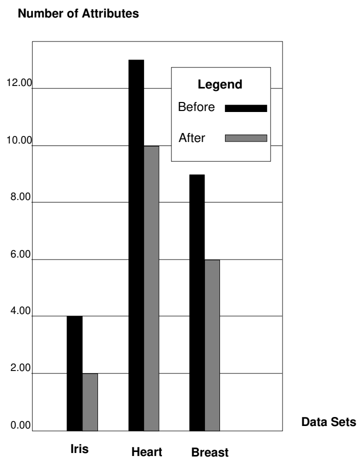

Figure 1: Number of attributes: original vs. those after Chi2 processing.

for the breast cancer and heart disease data. As for the Iris data, accuracy and tree size remain the same by using two attributes only (with 4 values each). In a way, it shows that C4.5 works pretty well without Chi2 for this data set. However, as the computational complexity of the procedure at each test node to compute information gain in C4.5 is $O(|C||A|)$ where $|C|$ is the number of patterns at the test node and $|A|$ is the number of attributes [9], after reducing the number of attributes (because of feature selection) and the number of values (because of discretization), the saving on time is obvious.

3.4 Empirical results on controlled data

The purpose of experimenting on the controlled data is to verify how effective Chi2 is in removing irrelevant attributes through discretizing numeric attributes. Therefore, it is only necessary to see if Chi2 can (1) discretize the relevant attribute(s) properly and (2) remove the irrelevant attributes.

Chi2 merged A_0 into three discrete values (1,2 and 3) corresponding to three classes (1,2, and 3); merged the other three attributes A_1, A_2, and A_3 into one value. That means that only A_0 should remain, and the noise (irrelevant) attributes should be removed.

For the modified Iris data, Chi2 merged six attributes out of eight. They are attributes 0, 1, 4, 5, 6 and 7. The first two are sepal-length and sepal-

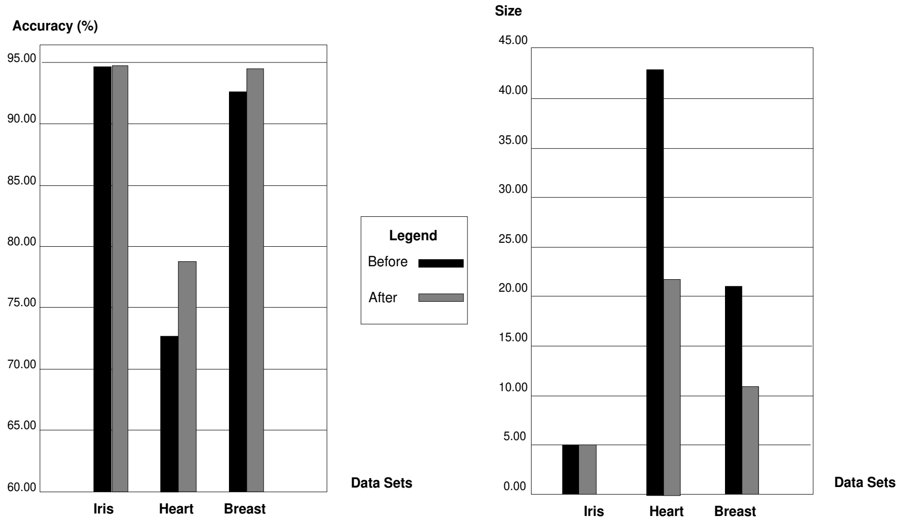

Figure 2: (a) Predictive accuracy and (b) size of decision trees of C4.5 for the three data sets after and before the Chi2 processing.

width. The last four are added noise (irrelevant) attributes. The remaining two attributes have been merged into 4 discrete values respectively as did in the real data experiment.

Through this set of controlled experiments, it is shown that Chi2 effectively discretizes numeric attributes and removes irrelevant attributes.

4 Discussions

4.1 Significant features of Chi2

This work is inspired by Kerber's ChiMerge [6]. ChiMerge requires a user to specify a proper significance level (α) which is used for merging values of all the attributes. No definite rule is given to choose this α. In other words, it is still a matter of try-and-error, and clearly it is not easy to find a proper significance level for each problem. Phase 1 of Chi2 extends ChiMerge to an automated one. That is α is automatically varied until the further merging is discontinued by the stopping criterion (the inconsistency rate). What makes Chi2 special is its capability of feature selection - a big step forward from discretization. In Phase 2 of Chi2, each attribute has its own significance level for merging in a round robin fashion. Merging stops when the inconsistency rate exceeds a specified one δ. This phase of Chi2 accomplishes feature selection. Another feature of Chi2 is that it can be applied to data with mixed attributes (e.g., Heart Disease Data). In addition, Chi2 can work with multi-calss data. This is an advantage over some statitic-based feature selection algorithms such as Relief [7] which is applicable only to the two-class data.

4.2 Selecting δ

The only threshold required is the tolerable rate of inconsistency, δ. Its default value is 0 assuming that the data set is consistent, and can be reset to any value between 0 and 1. A reasonable approximation is the rate of inconsistency found in the training data, which is not difficult to compute. With this threshold, Chi2 can guarantee that the dimensionally reduced data set will not have more inconsistencies than in the original one.

4.3 Computational complexity

Since each *Merge* in the Chi2 algorithm only reduces the number of intervals by one, in the worst case (there are n different values and all of them can be merged into one value), the innermost loop requires $n-1$ times of calling the χ^2 function (n is the number of patterns in the training data[3]), but each *Sort* needs $O(n \log n)$. So for m attributes, the reimplemented ChiMerge requires $O(mn \log n)$. Consider the worst case, checking data *Consistent* or not (refer to the Chi2 algorithm on page 4) takes $O(mn^2)$. The outermost loop is determined by the number of incremental steps, K, of the χ^2 value. Hence, the computational complexity of Phase 1 is $O(Km(n^2 + n \log n))$, i.e., $O(Kmn^2)$. Similar complexity can be obtained for Phase 2. The complexity result gives

[3] n is also the upper limit of the number of values an attribute can take in the sample data.

a guideline on how long it would take to run Chi2 for a given data set.

4.4 Limitations

Chi2 can only be used to discretize data and select features for supervised learning tasks since class information is vital in the χ^2 statistic. Also, Chi2 works on ordinal attributes only. If there are mixed (nominal and ordinal) attributes, Chi2 can be specified to operate only on the ordinal attributes for discretization and feature selection. Other feature selection algorithms for discrete data can be employed later to perform further feature selection. For example, the information gain measure can be used to select features [8]. Chi2 is only attempting to discover first-order (single attribute) correlations, thus might not perform correctly when there is a second-order correlation without a corresponding first-order correlation.

4.5 Other applications of Chi2

Because Chi2 discretizes a numeric attribute based on its statistics, it can be used to remove bounded inconsistency [5] by letting $\delta > 0$. An example of bounded inconsistency given by Hirsh is as follows: consider a pattern that is classified as positive, with an attribute whose value is 5.0 (measured by the real world device). If the tolerance on the measurement of the attribute's value is 0.1, the pattern could really have been one with feature value 4.9. If the "true" positive pattern were 4.9, and the pattern that really has value 5.0 would have been negative.

Another use of Chi2 is to extract production rules from simple data such as Iris [4]. This is a by-product of applying Chi2 to discretizing the training data. Chi2 is used to prescreen the data. If the data set processed by Chi2 requires only a few attributes, each with a few discrete values, the rules may be extracted with little effort. For example, the Iris data has only six distinct patterns (out of 75 training patterns) left (described by 2 attributes, 4 values each) after Chi2. Three rules obtained are very much the same as the first three rules generated by C4.5rules except for its default rule. However, this does not mean that Chi2 can replace classification algorithms. In general, Chi2 should be mainly served as a tool for feature selection and discretization of numeric attributes.

5 Conclusion

Chi2 is a simple and general algorithm that can automatically select a proper χ^2 value, determine the intervals of a numeric attribute, as well as select features according to the characteristics of the data. It guarantees that the fidelity of the training data can remain after Chi2 is applied. The empirical results on both the real data and controlled data have shown that Chi2 is a useful and reliable tool for discretization and feature selection of numeric attributes.

References

[1] H. Almuallim and T.G. Dietterich. Learning Boolean concepts in the presence of many irrelevant features. *Artificial Intelligence*, 69(1-2):279–305, November 1994.

[2] J. Catlett. On changing continuous attributes into ordered discrete attributes. In *European Working Session on Learning*, 1991.

[3] U.M. Fayyad and K.B. Irani. The attribute selection problem in decision tree generation. In *AAAI-92, Proceedings Ninth National Conference on Artificial Intelligence*, pages 104–110. AAAI Press/The MIT Press, 1992.

[4] R.A. Fisher. The use of multiple measurements in taxonomic problems. *Ann. Eugenics*, 7(2):179–188, 1936.

[5] H. Hirsh. Generalizing version spaces. *Machine Learning*, 17(1):5–46, October 1994.

[6] R. Kerber. ChiMerge: Discretization of numeric attributes. In *AAAI-92, Proceedings Ninth National Conference on Artificial Intelligence*, pages 123–128. AAAI Press/The MIT Press, 1992.

[7] K. Kira and L.A. Rendell. The feature selection problem: Traditional methods and a new algorithm. In *AAAI-92, Proceedings Ninth National Conference on Artificial Intelligence*, pages 129–134. AAAI Press/The MIT Press, 1992.

[8] H. Liu and W.X. Wen. Concept learning through feature selection. In *Proceedings of First Australian and New Zealand Conference on Intelligent Information Systems*, 1993.

[9] J.R. Quinlan. Induction of decision trees. *Machine Learning*, 1(1):81–106, 1986.

[10] J.R. Quinlan. *C4.5: Programs for Machine Learning.* Morgan Kaufmann, 1993.

[11] H. Ragavan and L. Rendell. Lookahead feature construction for learning hard concepts. In *Machine Learning: Proceedings of the Seventh International Conference*, pages 252–259. Morgan Kaufmann Pub. San Mateo, California, 1993.

[12] T.D. Ross, M.J. Noviskey, M.L. Axtell, D.A. Gadd, and J.A. Goldman. Pattern theoretic feature extraction and constructive induction. In *ACM ML/COLT'94 Workshop on Constructive Induction and Change of Representation*, 1994.

[13] N. Wyse, R. Dubes, and A.K. Jain. A critical evaluation of intrinsic dimensionality algorithms. In E.S. Gelsema and Kanal L.N., editors, *Pattern Recognition in Practice*, pages 415–425. Morgan Kaufmann Publishers, Inc., 1980.

Appendix

The appendix shows Chi2's behavior after Phase 1 and Phase 2 for the three attributes of the Iris data: sepal-width, petal-length, and petal-width. The tables summarize each attribute's intervals, class frequencies, and χ^2 values. All the tables are generated based on the 75 training data. Certainly, if all the 150 patterns are used, the tables could be different. Since the task of Chi2 is to facilitate learning, we assume that the testing data is not available at the preprocessing stage.

Int	Class Freq			χ^2
2.0	0	4	0	4.90
2.5	0	8	12	8.67
2.9	1	5	0	5.80
3.0	6	8	11	6.14
3.4	5	0	2	4.23
3.5	13	0	0	

(a)

Int	Class Freq			χ^2
2.0	25	25	25	

(b)

Table 3: The intervals, class frequencies, and χ^2 values for attribute sepal-width, after Phase 1 and Phase 2. The χ^2 thresholds are (a) 3.22 and (b) 40.6, respectively.

Int	Class Freq			χ^2
1.0	25	0	0	47.00
3.0	0	21	1	4.18
4.8	0	4	2	17.21
5.0	0	0	22	

(a)

Int	Class Freq			χ^2
1.0	25	0	0	53.00
3.0	0	25	3	39.39
5.0	0	0	22	

(b)

Table 4: The intervals, class frequencies, and χ^2 values for attribute petal-length, after Phase 1 and Phase 2. The χ^2 thresholds are (a) 3.22 and (b) 10.6, respectively.

Int	Class Freq			χ^2
0.1	25	0	0	38.10
1.0	0	13	0	4.72
1.4	0	2	1	3.37
1.5	0	9	0	11.35
1.7	0	1	5	3.40
1.9	0	0	19	

(a)

Int	Class Freq			χ^2
0.1	25	0	0	50.00
1.0	0	24	1	42.42
1.7	0	1	24	

(b)

Table 5: The intervals, class frequencies, and χ^2 values for attribute petal-width, after Phase 1 and Phase 2. The χ^2 thresholds are (a) 3.22 and (b) 10.6, respectively.