Frequency Count Based Filter for Dimensionality Reduction

B. Nath
Dept of Computer Science and Engineering
Tezpur University
Tezpur-784028, India
Email: bnath@tezu.ernet.in

D K Bhattacharyya
Dept of Computer Science and Engineering
Tezpur University
Tezpur-784028, India
E-mail:,dkb@tezu.ernet.in

A Ghosh
M IU and Center for Soft Computing Research
Indian Statistical Institute,
203 B. T. Road,
Kolkata- 700 108, India
ash@isical.ac.in

Abstract

Selecting relevant features from a dataset has been considered to be one of the major components of Data Mining techniques [1] [2]. Data mining techniques become computationally expensive when used with irrelevant features. Dimensionality reduction/feature selection algorithms are used basically to reduce the dimension of a dataset without reducing the information content of the domain. There are basically two categories of feature selection methods. Supervised, where each instance is associated with a class label, and in un-supervised, instances are not related to any class label. Un-supervised feature selection is used as a pre-processing of other machine learning techniques such as clustering, classification, association rule mining to reduce the dimensionality of the domain space without much loss of information content. This paper presents an un-supervised dimensionality reduction technique from continuous valued dataset, based on frequency count.

Keywords: Feature selection, association rule, frequency count, comprehensibility.

1. Introduction

Almost every dataset contains some irrelevant features. That is why removing the irrelevant features from the datasets have been a major research area for several decades. In reality, relevant features are unknown apriori. Therefore, many candidate features are introduced to represent the domain better way. It has been found from the experiments that many of the features are either irrelevant or redundant to the target concept. A relevant feature is neither irrelevant nor redundant to the target concept and an irrelevant feature does not affect the target concept in any way, and a redundant feature does add anything new to the target concept [3]. In many applications, the size of the dataset is so huge that learning might not work well before removing the unwanted features. Reducing the number of irrelevant or redundant features drastically reduces the execution time of a learning algorithm [4], [5].

Dimensionality reduction attempts to remove irrelevant features according to two basic criteria: (i) the accuracy does not significantly decrease and (ii) the resulting concept, given only the values for the selected features, is as close as possible to the original concept, given all the features. The dimensionality reduction methods find the best feature subset in terms of some evaluating function among the possible 2^N(where, N is the number of features) subsets.

A good number of algorithms had been proposed for dimensionality reduction over the years [6]. Some of the prominent feature selection algorithms are Branch & Bound [7], Focus [8], Relief [9], LVF [10], etc. This paper presents an algorithm for dimensionality reduction from continuous valued dataset based on frequency count. This paper exploits the concept of frequency count to extract the relevant features.

The rest of the paper is organized as follows: Section 2 describes some of the popular algorithms for feature selection; section 3 presents the proposed algorithm. Finally, Section 4 gives some experimental results to establish that the proposed algorithm is good enough to reduce the dimensionality of dataset to be used by a learning algorithm.

2. Existing Feature Selection Algorithms

In this section, some of the popular dimensionality reduction algorithms are reproduced. The notations/symbols used in describing those algorithms are reported in Table 1.

2.1. LVF [10]

Using consistency measure to evaluate the subsets, LVF generates the candidate subsets randomly. It randomly searches the subset space and calculates an inconsistency count for the subset. An inconsistency threshold is assumed and any subset with inconsistency measure greater than that value is rejected. The algorithm is given below.

LVF(D, S, MaxTries, λ)

1. T = S
2. For i=1 to *MaxTries*
3. Randomly choose a subset of features, S_j
4. if card(S_j) $\leq$ card(T)
5. if inConCal(S_j , D) $\leq \lambda$ then T = S_j and Output S_j
6. else append S_j to T
7. output S_j as 'another solution'
8. endfor
9. return T

Fig. 1. LVF

2.2. Branch and Bound [7]

This is an exponential search algorithm and was proposed by Narendra and Fukonaga in 1977. The important requirement of the algorithm is that the evaluation function be monotonic. The algorithm needs input of required number of features (M) and it attempts to find out the best subset. The algorithm is given below.

B&B(D, S,M)

1. if card(S) $\neq$ M then /*subset generation*/
2. j=0
3. for all features f in S begin
4. S_j = S - f /*remove one feature at a time */
5. if (S_j is legitimate) and if isbetter(S_j , T) then T = S_j
6. B&B(S_j ,M) /*recursion*/
7. j ++
8. return T

Fig. 2. Branch & Bound

2.3. Relief [9]

This algorithm selects the relevant features by using statistical method. It is basically a feature weight based algorithm designed based on instance based learning algorithm [11]. It first chooses a sample of instances (where the number of instances i.e. *Nosample* is a user input) at random from the set of training instances and for each instance in it, finds the *NearHit* and *NearMiss* instances based on Euclidian distance measure. *NearHit* of an instance is defined as the instance having minimum Euclidean distance among all instances of the same class as that of the instance. *NearMiss* of an instance is defined as the instance having minimum Euclidean distance among all instances of different class. The algorithm finds the weights of the features from a sample of instances and chooses the features with weight greater than a threshold. The algorithm is given below.

Relief(D, S, NoSample, Threshold)

1. T = Φ
2. Initialize all weights,W_j to zero
3. For i = 1 to *NoSample*
4. Randomly choose an instance x in D
5. find its *nearHit* and *nearMiss*
6. For j = 1 to N /* N is the number of features*/
7. $W_j = W_j - diff(x_j , nearHit_j)^2 + diff(x_j , nearMiss_j)^2$
8. For j =1 to N
9. If $W_j \geq$ *Threshold* , append feature f_j to T
11. Return T

Fig. 3. Relief

Relief works for noisy and correlated features. It can not work with redundant features and hence generates non-optimal features if the database contains redundant features. It works only with binary classes. Another problem is to choose the proper value of *NoSample*.

This algorithm is efficient as only the subset having the number of features smaller than that of the current best subsets are checked for inconsistency. It is easy to implement and finds the optimal subsets for most of the datasets.

2.4. DTM [12]

Decision Tree Method uses feature selection in an application on Natural Language Processing. To select the features, it runs C4.5 [13] over a training set and all those features that appear in the pruned decision tree are selected. In other words, the union of the subsets of the features, appearing in the path to any leaf node in the pruned tree, is the selected subset.

2.5. MDLM [14]

Minimum Description Length Method tries to eliminate all irrelevant and redundant features. This method is based on the concept that if the features in a subset X can be expressed as a fixed non-class-dependent function F of the features in another subset Y, then once the values in the features in the subset X are known, the features in the subset Y are useless. Minimum Description Length Criterion (MDLC) is used for this purpose. The algorithm exhaustively searches all the possible subsets and returns the subset satisfying MDLC. This method can find all the useful features for Gaussian cases.

Table 1. Symbols/Notations Used in the Algorithms

D	=	The Database
S	=	Original set of Features
M	=	Number of features to be selected
Card(X)	=	Function to find the cardinality of the set X
isbetter(X; Y)	=	A function to check if the set X is better than the set Y
NoSample	=	the sample size
ThresHold	=	lower limit of a feature's weight to become relevant
N	=	number of features
γ	=	Minimum support

Wj	=	weight of j-th feature
Maxtries	=	number of iterations
InConCal	=	function to calculate inconsistency
λ	=	upper level of inconsistency
diff()	=	to find difference of same feature in two different records
L_1	=	Features frequent occurrence
L_1	=	Features whose non occurrence is frequent
β	=	Increment to min-support
F,F1	=	Set of selected attributes

2.6. FFC[15]

Based on coherence properties of an attribute to the target concept, FFC tries to select the relevant itemsets. For selecting them it uses the coherence frequency count and non coherence frequency count of the attributes.

FFC(D, γ,β, n)
1. F=all the features
2. do while(|F| > n)
   3. S=ϕ , L_1 = {f | support(f)$\geq \gamma$} ,
   L_1'={f | support(f')$\geq \gamma$}
   4. S=S$\cup${xC | x$\in$ $L_1 \cup L_1'$
   5. for all instances i $\in$ D do begin
      6. S_i=subset(S, i)
      7. for all s $\in$ S_i do
         8. s.count++
   9. F1 = F
   10. F = {f | s = fC , s $\in$ S_i and s.count $\geq \gamma$}
   11. γ=γ + β
12. if |F| = n then return F
13. else return F1

Fig. 4. FFC

3. The Proposed Algorithm

The proposed algorithm DRUFT(**D**imensionality **R**eduction **U**sing **F**requency coun**T**) is meant for reducing the dimensionality of market basket dataset based on Frequency count. The above motioned algorithms also reduce the dimensionality of the dataset by selecting only the relevant attributes from the original continuous valued dataset. When the dataset is converted to market basket, all the sub-ranges of the selected attributes have to be considered. But some sub-ranges of those selected attributes may again be irrelevant. If these sub-ranges are also can be eliminated then the dimensionality will be further reduced. Unlike the above mentioned algorithms, DRUFT is capable of finding the relevant sub-ranges of the attributes, resulting in a market basket dataset with a few numbers of attributes in it.

The algorithm takes the dataset and maximum number of needed sub-range, i.e. *Maxatt*, as input. *Table 2* describes the symbols used in DRUFT. It reads the dataset only once and finds the frequency count of every sub-range of all attributes. Using these frequency counts it eliminates irrelevant sub-ranges, till desired number of attributes remains not-eliminated. Finally it produces the not-eliminated sub-ranges as output. Number of such sub-ranges will be equal or less than *MaxAtt*.

DRUFT(D, *MaxAtt*)
1. S=ϕ
2. for all attributes $A_i \in A$
   3. for all subranges $P_{i,j} \in A_i$
      4. S=S $\cup$ $P_{i,j}$
5. for all s $\in$ S
   6. find the frequency count, SUP_s
7. *minfreq*=1
8. . S1=ϕ
9. for all s $\in$ S
   10.if (s $\in$ A_i)and ((SUP_s * |A_i|) $\geq$ (*minfreq* * max(|A|))
      11. S1=S1$\cup$s
12. if |S1| $\leq$ *MaxAtt* go to step 16
13. *minfreq*= *minfreq*+1
14. S=S1
15. goto step 8
16. return S1

Fig. 5. DRUFT

The proposed algorithm works on the original continuous valued dataset where the number of attributes is very small hence requiring less amount of

memory for its execution. For every attribute some sub-ranges are considered. These sub-ranges will become attributes in the market basket dataset. But the proposed method will restrict some of these ranges from becoming an attribute of the market basket dataset. For every sub-range of all the attributes, the frequency of them within the dataset is calculated by reading the dataset once. Afterwards, only those frequency counts will be used to reduce the dimensionality of the market basket dataset to the user desired level. The user has to provide his desired number of attributes as input to the algorithm. After calculating the frequency count, those sub-ranges are eliminated; whose frequency count is less than a factor of minimum frequency, *minfreq*. This factor is different for the sub-ranges of different attribute. If the dataset has been reduced to the desired level, it will produce the sub-ranges that are found out to be relevant. Otherwise it will eliminate some more sub-ranges by incrementing the minimum frequency count, *minfreq*, This process will continue till the number of relevant sub-ranges do not become less than or equal to the user desired number of attributes.

Table 2: symbols used in DRUFT

A	=	Attributes of original dataset
$\|A_i\|$	=	Number or sub-ranges of i^{th} Attribute
$P_{i,j}$	=	j^{th} sub-range of i^{th} attribute
max(|A|)	=	Maximum of $\|A_i\|$
S	=	Set of sub-ranges of A
S1	=	Set of selected sub-ranges
SUP_s	=	Frequency of sub range s
minfreq	=	Current value of support count to declare as frequent
MaxAtt	=	Maximum no of sub-ranges to be selected

4.1. Discussion

This algorithm uses the concept of frequency count of sub-range of values of the attribute. For elimination it will use a minimum support count that starts from 1 and increments it at every iteration. Some of the above mentioned algorithms use it as a user parameter. So it affects the output of the algorithm, but DRUFT is free from it. The execution time of the algorithm is controlled only by the desired number of attributes, as it is the only user parameter. It always attempts to reduce the dimensionality irrespective of the size of the dataset.

4. Experimental Results

The proposed algorithm was tested with several test datasets as well as *Monks-1* and *Monks-3* training datasets downloaded from UCI machine learning repository. Results for the later two datasets are analyzed below. There are 124 and 122 instances in Monks-1 and Monks-3 respectively. Both of them have 8 attributes; first one is the class number and the last one is the sample number. Remaining six attributes are numeric values spanning over different ranges. The minimum and maximum values of these attributes are A1(1,3), A2(1,3), A3(1,2), A4(1,3),A5(1,4) and A6(1,2). If these datasets are converted to market basket then there will be total 17 attributes. From the following results it can be observed that it selects the sub-ranges of the attributes those were declared as relevant by the existing algorithms. Only part 2 of attribute 6, denoted in the Table 3 as A6-2, is coming in addition. Reason for not selecting A6 by other algorithm is that it is a redundant attribute. Some results of the proposed algorithm on monks datasets are listed in Table 3; Table 4 compares results of DRUFT with some of the existing algorithms.

5. Conclusions

This paper has presented an algorithm for dimensionality reduction in continuous valued data based on frequency count. Based on experimentation, it has been found that the proposed algorithm is good enough for reducing the dimension of market basket dataset.

Table3. Dimensionality Reduction on Monks-1 and Monks-3 by DRUFT

	Monks-3		Monks-1	
Desired no attributes	Reduced to	Minimum support	Reduced to	Minimum support
10	8	31	10	31
9	8	31	6	32
8	8	31	6	32
7	4	32	6	32
6	4	32	6	32
5	4	32	4	33

Table 4. Comparative results of some existing algorithms and DRUFT

Method	Monk3		Monk1	
	Selected attributes	MB's dimension	Selected attributes	MB's dimension
Relief	A2,A5 always& one or both of A3,A4	9 or 10 or 12	A1,A2,A5	10
B&B	A1,A3,A4	8	NA	-
DTM	A2,A5	7	NA	-
LVF	A2,A4,A5	10	NA	-
MDLM	A2,A3,A5	9	NA	-
FFC	A1,A2,A4,A5	13	A1,A2,A5	10
DRUFT Reduced to 4	A1-1,A3-1,A4-3,A5-1	4	A1-1,A2-3,A5-4,A6-2	4
DRUFT Reduced to 6	-	-	A1-1,A2-3, A3-1, A4-3, A5-4,A6-2	6
DRUFT Reduced to 8	A1-1,A2-2,A3-1,A4-3, A5-1,A5-2,A5-4,A6-2	8	-	-

References

[1] Agarwal R. and Shrikant R., Fast algorithms for mining association rules, in 20th VLDB Conf, Sept. 1994.

[2] Agarwal R., Mannila H., Shrikant R., Toivonan H. and Verkamo A .I, Fast discovery of association Rules, in U. Fayyad and et al, editors Advances in Knowledge Discovery and Data mining, MIT Press,1996.

[3] Kohavi R and Peger K, Irrelevant Features and Subset Selection Problem. In Proceedings of the Eleventh International Conference on Machine Learning, 121-129,1994.

[4] Kohavi R and Sommerfield D, Feature Subset Selection using the Wrapper Method: Over fitting and dynamic search space topology. In Proceedings of First International Conference on Knowledge Discovery and Data Mining, Morgan Kaufman, 192-197,1995

[5] Koller D and Sahami M, Towards Optimal Feature Selection. In Proceedings of International Conference on Machine Learning,1996.

[6] Doak J, An evaluation of feature selection methods and their applicaton to computer security. Technical report, Davis, CA: University of Califomia, Department of Computer Science,1992.

[7] Narendra P M and Fukunaga K, A branch and bound algorithm for feature selection. IEEE Transaction on Computers,C-26(9):917-922,September,1977.

[8] Almuallim H and Dietterich,T G, Learning with many Irrelevant Features. In Proceedings of Ninth National Conference On Artificial Intelligence, MIT press, Cambridge, Massachusetts,547-552,1992.

[9] Kira K and Rendell L A, The Feature selection problem: Traditional Methods and a New Algorithm. In Proceedings of Ninth National Conference on Artificial Intelligence, L9-134,1992.

[10] Liu H and Setiono R, A Probabilistic Approach to Feature Selection -a filter solution. In Proceedings of International Conference on Machine Learning,319-327,1996.

[11] Dash, M and Liu, H., Feature Selection for Classification, Intelligent Data Analysis, vol 1, pp. 131-156, 1997.

[12] Cardie C. Using decision trees to improve case based learning. In Proceedings of Tenth International Conference on Machine Learning, 1994.

[13] Quinlan J R. C4.5: Programs for Machine Learning. Morgan Kaufman, San Matco, California, 1993.

[14] Shieinvald J, Dom B and Niblack W. A modelling approach to feature selection. In Proceedings of Tenth International Conference on Pattern Recognition,1:535-539, June 1990.

[15] A Das and D K Bhattacharyya, FFC:Feature Selection Using Frequency Count, In the proceedings of ICISIP, Chennai,2005.