Search-based Class Discretization

Luís Torgo
email : ltorgo@ncc.up.pt

João Gama
email : jgama@ncc.up.pt

LIACC - University of Porto
R. Campo Alegre, 823 - 4150 Porto - Portugal
Phone : (+351) 2 6001672 Fax : (+351) 2 6003654
WWW : http://www.ncc.up.pt/liacc/ML

Abstract. We present a methodology that enables the use of classification algorithms on regression tasks. We implement this method in system RECLA that transforms a regression problem into a classification one and then uses an existent classification system to solve this new problem. The transformation consists of mapping a continuous variable into an ordinal variable by grouping its values into an appropriate set of intervals. We use misclassification costs as a means to reflect the implicit ordering among the ordinal values of the new variable. We describe a set of alternative discretization methods and, based on our experimental results, justify the need for a search-based approach to choose the best method. Our experimental results confirm the validity of our search-based approach to class discretization, and reveal the accuracy benefits of adding misclassification costs.

Keywords : Regression, Classification, Discretization methods.

1 Introduction

Machine learning (ML) researchers have traditionally concentrated their efforts on classification problems. However, many interesting real world domains demand for regression tools. In this paper we present and evaluate a discretization method that extends the applicability of existing classification systems to regression domains. The discretization of the target variable values provides a different granularity of predictions that can be considered more comprehensible. In effect, it is a common practice in statistical data analysis to group the observed values of a continuous variable into class intervals and work with this grouped data (Bhattacharyya & Johnson, 1977). The choice of these intervals is a critical issue as too many intervals impair the comprehensibility of the models and too few hide important features of the variable distribution. The methods we propose provide means to automatically find the optimal number and width of these intervals. The motivation for transforming regression into classification is to obtain a different tradeoff between comprehensibility and accuracy of regression models. As a by-product of our methods we also broaden the applicability of classification systems.

We argue that mapping regression into classification is a two-step process. First we have to transform the observed values of the goal variable into a set of intervals. These intervals may be considered values of an ordinal variable (i.e. discrete values with an implicit ordering among them). Classification systems deal with discrete target variables. They are not able to take advantage of the given ordering. We

propose a second step whose objective is to overcome this difficulty. We use misclassification costs which are carefully chosen to reflect the ordering of the intervals as a means to compensate for the information loss regarding the ordering.

We describe several alternative ways of transforming a set of continuous values into a set of intervals. Initial experiments revealed that there was no clear winner among them. This fact lead us to try a search-based approach to this task of finding an adequate set of intervals. We use a wrapper technique (John et al., 1994; Kohavi, 1995) as a method for finding near-optimal settings for this mapping task.

We have tested our methodology on four regression domains with three different classification systems : C4.5 (Quinlan, 1993); CN2 (Clark & Nibblet, 1988); and a linear discriminant (Fisher, 1936; Dillon & Goldstein, 1984). The results show the validity of our search-based approach and the gains in accuracy obtained by adding misclassification costs to classification algorithms.

The next section describes how to transform a continuous target variable into a set of intervals. In section 3 we describe our proposal of using misclassification costs to deal with ordinal variables. The experiments we carried out done are described on section 4. Finally, we comment the relations to other work and present the conclusions of this paper.

2 Obtaining a Set of Intervals

In regression problems we are given samples of a set of independent (predictor) variables $x_1, x_2, \ldots, x_n$, and the value of the respective dependent (output) variable y. Our goal is to obtain a model that captures the mapping $y = f(x_1, x_2, \ldots, x_n)$ based on the given samples. Classification differs from this setup in that y is a discrete variable instead of continuous one.

Mapping regression into classification can be seen as a kind of pre-processing technique that enables the use of classification algorithms on regression problems. Our method starts by creating a data set with discrete target variable values. This step involves examining the original continuous values of the target variable and suitably dividing them into a series of intervals. Every example whose output variable value lies within the boundaries of an interval will be assigned the respective "discrete class"[1] .

Grouping the range of observed continuous values of the target variable into a set of intervals involves two main decisions : how many intervals to create; and how to choose the interval boundaries. As for this later issue we use three methods that for a given a set of continuous values and the number of intervals return their defining boundaries :

- *Equally probable intervals (EP):* This creates a set of N intervals with the same number of elements.
- *Equal width intervals (EW):* The original range of values is divided into N intervals with equal width.

---

[1] We use the median of the values lying within the interval as class label.

- *K-means clustering (KM):* In this method the aim is to build N intervals that minimize the sum of the distances of each element of an interval to its *gravity center* (Dillon & Goldstein, 1984). This method starts with the EW approximation and then moves the elements of each interval to contiguous intervals whenever these changes reduce the referred sum of distances.

The number of intervals used (i.e. the number of classes) will have a direct effect on the accuracy of the subsequent learned theory. This means that they can be seen as a parameter of the learning algorithm. Our goal is to set the value of this "parameter" such that the system performance is optimized. As the number of possible ways of dividing a set of continuous values into a set of intervals is potentially infinite a search algorithm is necessary. The *wrapper* approach (John et al., 1994; Kohavi, 1995) is a well known strategy has been mainly used for feature subset selection (John et al., 1994) and parameter estimation (Kohavi, 1995). The use of this iterative approach to estimate a parameter of a learning algorithm can be described by the following figure:

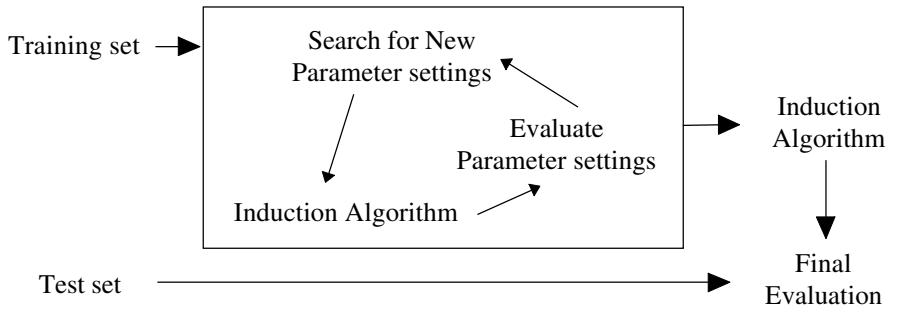

Figure 1 - The wrapper approach.

The two main components of the wrapper approach are the way new parameter settings are generated and how their results are evaluated in the context of the target learning algorithm. The basic idea is to try different parameter settings and choose the one that gives the best estimated accuracy. This best setting found by the wrapper process will then be used in the learning algorithm in the *real* evaluation using an independent test set. As for the search component we have used a hill-climbing search coupled with a settable lookahead parameter to minimize the well-known problem of local minima. Given a tentative solution and the respective evaluation the search component is responsible for generating a new tentative. We provide the following two alternative search operators :

- *Varying the number of intervals (VNI):* This simple alternative consists of incrementing the previously tried number of intervals by a constant value.
- *Incrementally improving the number of intervals (INI) :* The idea of this alternative is to try to improve the previous set of intervals taking into account their individual evaluation. The next set of intervals is built using the median of

these individual error estimates. All intervals whose error is above the median are further split by dividing them in two. All the other intervals remain unchanged.

These two alternatives together with the three given splitting strategies make six alternative discretization methods which can be tuned to the given task using a wrapper approach. The RECLA system allows the user to explicitly select one of these methods. If this is not done the system automatically selects the one that gives better estimated results.

The other important component of the wrapper approach is the evaluation strategy. We use a N-fold Cross Validation (Stone, 1974) estimation technique which is well-known for its reliable estimates of prediction error. This means that each time a new tentative set of intervals is generated RECLA uses an internal N-fold Cross Validation (CV) process to evaluate it. In the next subsection we provide a small example of a discretization process to better illustrate our search-based approach.

2.1 An illustrative example

In this example we use the *servo* data set (see details in section 4). We have coupled RECLA with C4.5 and evaluated the learned model with the MAE statistic (see section 3). We have performed two experiments with different discretization methods. In the first experiment we use the VNI search operator with the KM splitting algorithm. Table 1 presents the discretizations of this first experiment (KM+VNI). The first column shows the number of intervals tried in each iteration[2] . The second column shows the obtained intervals by the KM splitting method. The second line of this column includes the median of the values within the intervals (the used "classes"). The last column gives the internal 5-fold CV error estimate of each tentative set of intervals. In this example we have used the value 1 for the "Lookahead" parameter mentioned before. The solution of this method is thus 4 intervals (the trial with best estimated error).

N.Ints	*Intervals / Discrete Class Values*						*Error*
2	[0.13-2.60]	[2.60-7.10]					*0.510*
	0.58	4.50					
4	**[0.13-0.45]**	**[0.45-0.75]**	**[0.75-3.20]**	**[3.20-7.10]**			**0.374**
	0.34	**0.54**	**1.03**	**4.50**			
6	[0.13-0.38]	[0.38-0.52]	[0.52-0.75]	[0.75-1.08]	[1.08-3.20]	[3.20-7.10]	*0.429*
	0.28	0.47	0.58	0.90	1.30	4.50	

Table 1 - Trace of KM+VNI method.

In the second experiment we use the same splitting algorithm but with the INI search operator. The results are given in Table 2. We also include the estimated error of each interval (the value in parenthesis). The next tried iteration is dependent on these estimates. The intervals whose error is greater than the median of the estimates are split in two intervals. For instance, in the third iteration we can observe that the third interval was maintained from the second trial, while the other were obtained by splitting a previous interval.

---

[2] The fact that starts with 2 and goes in increments of 2 is just an adjustable parameter of RECLA.

N.Ints	*Intervals / Discrete Class Values*					*Error*
1	[0.13-7.10]					*0.91*
	0.73 (0.91)					
2	[0.13-2.60]	[2.60-7.10]				*0.42*
	0.58 (*0.41*)	4.50 (*0.32*)				
3	**[0.13-0.73]**	**[0.73-2.60]**	**[2.60-7.10]**			**0.35**
	0.5 (*0.35*)	**1.90 (*0.35*)**	**4.50 (*0.33*)**			
5	[0.13-0.4]	[0.46-0.73]	[0.73-1.07]	[1.1-2.60]	[2.60-7.10]	*0.41*
	0.35 (*0.29*)	0.54 (*0.38*)	0.9 (*0.51*)	1.4 (*0.49*)	4.50 (*0.4*)	

Table 2 - Trace of KM+INI method.

The two methods follow different strategies for grouping the values. In this example the second alternative lead to lower error estimates and consequently this alternative was preferred by RECLA.

An interesting effect of increasing the number of intervals is that after some threshold the algorithm's performance decreases. This may be caused by the decrease of the number of cases per class leading to unreliable estimates due to overfitting the data.

3 Using Misclassification Costs with Ordinal Variables

Classification systems search for theories that have minimal estimated prediction error according to a 0/1 loss function, thus making all errors equally important. In regression, prediction error is a function of the difference between the observed and predicted values (i.e. errors are metric). Accuracy in regression is dependent on the amplitude of the error. In our experiments we use the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE) as regression accuracy measures :

$$MAE = \frac{\sum |y_i - y_i'|}{N} \qquad MAPE = \frac{\sum |(y_i - y_i')/y_i| \times 100}{N} \qquad (1)$$

In order to differentiate among different errors our method incorporates misclassification costs in the prediction procedure. If we take $c_{i,j}$ as the cost of classifying a class *j* instance as being class *i*, and if we take $p(j|\mathbf{X})$ as the probability given by our classifier that instance **X** is of class *j*, we can take the task of classifying instance **X** as finding the class *i* that minimizes the expression

$$\sum_{j \in \{\text{classes}\}} c_{i,j}\, p(j|\mathrm{X}) \qquad (2)$$

Here we associate classes with intervals and we take as class labels the intervals medians. In our methodology we propose to estimate the cost of misclassifying two intervals by using the absolute difference between their representatives, i.e.

$$c_{i,j} = |\tilde{y}_i - \tilde{y}_j| \qquad (3)$$

where $\tilde{y}_i$ is the median of the values that where "discretized" into the interval *i*.

By proceeding this way we ensure that the system predictions minimize the expected absolute distance between the predicted and observed values.

4 Experimental Evaluation

We have carried out several experiments with four real world domains. These data sets were obtained from the UCI machine learning repository (Merz & Murphy, 1996). Some of the characteristics of the data sets used are summarized in Table 3.

Data Set	N. Examples	N. Attributes
housing	506	13 (13C)
servo	167	4 (4D)
auto-mpg	398	7 (4C+3D)
machine	209	6 (6D)

Table 3- The used data sets (C - continuous attribute; D - discrete attribute).

RECLA system was coupled with C4.5 (a decision tree learner), CN2 (a rule-based system) and Discrim (a linear discriminant). MAE and MAPE were used as regression accuracy measures. The error estimates presented in Table 4 are obtained by a 5-fold cross validation test[3] . We also include the standard deviation of the estimates and the discretization methodology chosen by RECLA.

DataSet	Algorithm	MAE	MAPE
Servo	Cn2	0.38 ± 0.13 (ep-INI)	36.0 ± 9.03 (ep-INI)
	C4.5	0.36 ± 0.06 (ep-INI)	32.6 ± 8.6 (ep-INI)
	Discrim	0.39 ± 0.10 (km-VNI)	37.5 ± 4.0 (km-INI)
Auto-Mpg	Cn2	3.0 ± 0.3 (km-INI)	13.8 ± 2.07 (km-INI)
	C4.5	3.1 ± 1.6 (ep-VNI)	13.1 ± 3.6 (ep-VNI)
	Discrim	2.6 ± 1.1 (ep-VNI)	11.5 ± 2.3 (ep-VNI)
Housing	Cn2	3.68 ± 0.5 (ew-INI)	18.09 ± 3.8 (km-INI)
	C4.5	4.2 ± 0.66 (ew-VNI)	21.9 ± 5.7 (km-VNI)
	Discrim	3.8 ± 1.14 (km-VNI)	18.0 ± 4.85 (ep-VNI)
Machine	Cn2	46.8 ± 11.8 (km-INI)	43.3 ± 9.2 (ew-INI)
	C4.5	47.1 ± 24.5 (ep-INI)	47.4 ± 8.5 (ep-VNI)
	Discrim	38.8 ± 23.7 (km-VNI)	42.0 ± 11.6 (km-VNI)

Table 4 - Best results of Classification algorithms using RECLA.

The observed variability of the chosen discretization method provides an empirical justification for our search-based approach. We can also notice that the best method is dependent not only on the domain but also on the used induction tool (and less frequently on the error statistic). This justifies the use of a wrapper approach that chooses the best number of intervals taking these factors into account.

In another set of experiments we have omitted the use of misclassification costs. This lead to a significant drop on regression accuracy in most of the setups thus providing empirical evidence of the value of adding misclassification costs. However, it should be mentioned that misclassification costs cannot be used with all classification systems. In effect, if the system is not able to output class probability

[3] Notice that this test is independent from the internal cross validation that is performed by RECLA to estimate the best discretization.

distributions when classifying unseen instances, we are not able to use misclassification costs. The consequence will probably be a lower accuracy as our experiments indicate.

RECLA provides means of using different types of classification systems in regression tasks. The regression models obtained by this methodological approach are in a way more comprehensible to the user as the predictions have higher granularity. However, the loss of detail due to the abstraction of continuous values into intervals has some consequences on regression accuracy. We have tried to find out this effect by obtaining the results of some "pure" regression tools on the same data sets using the same experimental methodology. Table 5 shows the results obtained by a regression tree similar to CART (Breiman et al., 1984), a 3-nearest neighbor algorithm (Fix & Hodges, 1951) and a standard linear regression method :

Dataset	Algorithm	MAE	MAPE
Servo	Regression tree	0.43 ± 0.4	34.0 ± 9.9
	3-NN	0.52 ± 0.11	57.1 ± 17.3
	Linear Regression	0.87 ± 0.07	104.5 ± 22.7
Auto-Mpg	Regression tree	2.6 ± 0.3	11.22 ± 0.9
	3-NN	2.4 ± 0.25	10.05 ± 1.5
	Linear Regression	2.5 ± 0.23	11.06 ± 0.9
Housing	Regression tree	2.8 ± 0.2	14.5 ± 2.2
	3-NN	2.9 ± 0.4	13.7 ± 1.08
	Linear Regression	3.4 ± 0.4	16.5 ± 2.7
Machine	Regression tree	46.8 ± 12.6	54.5 ± 9.47
	3-NN	34.1 ± 11.1	47.45 ± 8.8
	Linear Regression	36.8 ± 6.7	58.9 ± 14.6

Table 5 - Performance of Regression Tools.

These results are comparable with the ones given in Table 4. This means that our approach can provide an interesting alternative when a different trade-off between accuracy and comprehensibility is needed.

5 Related Work

Mapping regression into classification was first proposed by Weiss & Indurkhya (1993, 1995). These authors incorporate the mapping within their regression system. They use an algorithm called P-class that splits the continuous values into a set of K intervals, and use cross validation to estimate the number of intervals. Their methodology is similar to our KM+VNI discretization. Compared to this work we added other alternative discretization methods and empirically proved the necessity of a search-based approach to class discretization. Moreover, by separating the discretization process from the learning algorithm we extend this approach to other systems. Finally, we have introduced the use of misclassification costs to overcome the inadequacy of classification systems to deal with ordinal target variables.

Previous work on continuous attribute discretization usually proceeds by trying to maximize the mutual information between the resulting discrete attribute and the

classes (Fayyad & Irani, 1993). This strategy is applicable only when the classes are given. Ours is a different problem, as we are determining which classes to consider.

6 Conclusions

The method described in this paper enables the use of classification systems on regression tasks. The significance of this work is two-fold. First, we have managed to extend the applicability of a wide range of ML systems. Second, our methodology provides an alternative trade-off between regression accuracy and comprehensibility of the learned models. Our method also provides a better insight about the target variable by dividing its values in significant intervals, which extends our understanding of the domain.

We have presented a set of alternative discretization methods and demonstrated their validity through experimental evaluation. Moreover, we have added misclassifications costs which provide a better theoretical justification for using classification systems on regression tasks. We have used a search-based approach which is justified by our experimental results which show that the best discretization is often dependent on both the domain and the induction tool.

References

Breiman,L. , Friedman,J.H., Olshen,R.A. & Stone,C.J. (1984): *Classification and Regression Trees*, Wadsworth Int. Group, Belmont, California, USA, 1984.

Bhattacharyya,G., Johnson,R. (1977) : *Statistical Concepts and Methods*. John Wiley & Sons.

Clark, P. and Niblett, T. (1988) : The CN2 induction algorithm. In *Machine Learning*, **3**.

Dillon,W. and Goldstein,M. (1984) : *Multivariate Analysis*. John Wiley & Sons, Inc.

Fayyad, U.M., and Irani, K.B. (1993) : Multi-interval Discretization of Continuous-valued Attributes for Classification Learning. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence (IJCAI-93)*. Morgan Kaufmann Publishers.

Fisher, R.A. (1936) : The use of multiple measurements in taxonomic problems. *Annals of Eugenics*, **7**, 179-188.

Fix, E., Hodges, J.L. (1951) : Discriminatory analysis, nonparametric discrimination consistency properties. Technical Report 4, Randolph Field, TX: US Air Force, School of Aviation Medicine.

John,G.H., Kohavi,R. and Pfleger, K. (1994) : Irrelevant features and the subset selection problem. In *Proceedings of the 11th IML*. Morgan Kaufmann.

Kohavi, R. (1995) : Wrappers for performance enhancement and oblivious decision graphs. PhD Thesis.

Merz,C.J., Murphy,P.M. (1996) : UCI repository of machine learning databases [http://www.ics.uci.edu/MLReposiroty.html]. Irvine, CA. University of California, Department of Information and Computer Science.

Quinlan, J. R. (1993) : *C4.5 : programs for machine learning*. Morgan Kaufmann Publishers.

Stone, M. (1974) : Cross-validatory choice and assessment of statistical predictions. *Journal of the Royal Statistical Society*, B **36**, 111-147.

Weiss, S. and Indurkhya, N. (1993) : Rule-base Regression. In *Proceedings of the 13th International Joint Conference on Artificial Intelligence*, pp. 1072-1078.

Weiss, S. and Indurkhya, N. (1995) : Rule-based Machine Learning Methods for Functional Prediction. In Journal Of Artificial Intelligence Research (JAIR), volume 3, pp.383-403.