MARS: Generating and Reducing Conflicting Prototypes for Nearest Neighbour Classifiers with Application to Discrete Standard Datasets

Author:
Fayola PETERS

Supervisor:
Dr. Tim MENZIES

Word Count $\approx$ 2300
March 13, 2010

Abstract

Prototype Learning Schemes (PLS) started appearing in order to alleviate the drawbacks of nearest neighbour classifiers (NNC). Namely computation time, storage requirements, the effects of outliers on the classification results and also the negative effect of data sets with non-separable and/or overlapping classes. To that end all PLS have endeavored to create or select a *good* representation of training data which is a mere fraction of the size of the original training data. In most of the literature this fraction is about 10%. In this work, the design, implementation and evaluation of MARS, a new prototype learning scheme is described. MARS works by first generating initial-prototypes with a clustering algorithm. These prototypes are then examined for conflicts, i.e. prototypes in a local neighbourhood with conflicting classes. When or if these prototypes are found they are removed.

1 Introduction

Since the creation of the Nearest Neighbour algorithm in 1967 [CH67], various prototype learning schemes (PLS) have appeared to remedy the four (4) major drawbacks associated with the algorithm and it's variations. First, the high computation costs caused by the need for each test sample to find the distance between it and each training sample. Second, the storage requirement is large since the entire dataset needs to be stored in memory, third, outliers can negatively affect the accuracy of the classifer and fourth the negative effect of data sets with non-separable and/or overlapping classes. To solve these issues, PLS are used. Their main purpose is to reduce a training set via various selection and/or creation methods to produce *good prototypes*. *Good* here meaning that the prototypes are a good representation of the original training dataset such that they maintain comparable or increased classification accuracy of a nearest neighbour classifier.

Earlier successes with prototype generation include Chang's work in 1974 [Cha74], LVQ (learning vector quantization) [Koh90, KS98] and the condensed nearest neighbour (CNN) rule [Har68]. These early works form the basis of today's PLS, for instance, minimal consistent set (MCS) [Das94] by Darsarathy based on CNN. In Chang's algorithm, every point in the training set (T) starts out as a prototype. Then, in turn, any two(2) nearest prototypes (p1 and p2) of the same class are merged to form p* which replaces p1 and p2 and adopts their class label. This merging takes place if and only if there is no downgrade in the classification (Nearest Neighbour Classification -NNC) of instances in T. Also, merging is done with euclidean distance or weighted euclidean distance. Chang's merging process continues "...until the number of incorrect classifications of patterns in T starts to increase" [Cha74]. More than a decade after Chang's work, Kohonen [Koh90] in 1990, introduced the learning vector quantization (LVQ). LVQ is a set of learning algorithms for nearest prototype classification and its basic algorithm, LVQ1, works by first selecting a certain number of prototypes from each class randomly as initial prototypes. This ensures that each class is represented be at least one prototype. These initial prototypes are then updated using the training set with the basic idea that the prototypes will be attracted to training points with the same class label and repelled by those with different class labels. CNN is one of the oldest PLS. Introduced by Hart [Har68] in 1968, the algorithm works as follows:

> ...first a single pattern is put in the condensed set.Then each pattern is considered and its nearest neighbour in the condensed set is found. If its label is the same as that of the pattern in the condensed set, it is left out; otherwise the new pattern is included in the condensed set. After one pass through all the training patterns, another iteration is carried out where each training pattern is classified using the patterns in the condensed set. These iterations are carried out till every training pattern is correctly classified using the NNC on the patterns in the condensed set. [DM02]

Finally, looking at MCS [Das94], a more recent PLS, Darsarathy improves on CNN by abandoning the exhaustive search and ensuring that the final set of prototypes is a *minimal* and *consistent* set. Basically, Darsarthy's method selects a protoype for the set if it has the greater number of instances of similar class closer to it than the closest instance of a different class, i.e it nearest unlike neighbour (NUN).

Drawing from these previous efforts, particularly Darsarathy's MCS, this paper presents MARS (MESO[1] and Reduction Solution). MARS is a procedure which uses MESO [KM07], a perceptual memory system to generate *inital prototypes*. Briefly, in MESO prototypes generated are cluster based and are called spheres. Each sphere contains a collection of similar instances. Sphere membership is dictated by a grow function which manages the size of the sphere. So, any new instance finds its closest sphere, if the distance is less than the distance of the growth function then it gets added to the sphere otherwise a new sphere is created with the instance [KM07]. Once the initial prototypes are generated, a novel technique for reducing its number is applied. The goal of this novel reduction approach is to remove conflicting overlapping prototypes which can lead to the misclassification of test instances.

[1] Multi-Element Self-Organizing Tree

The remainder of this paper is organized as follows: Section 2 describes the MARS procedure; MESO's algorithm and how it is used to suit our purpose of creating prototypes, followed by a detailed description of reduction solution. Section 3 presents experimental results which evaluates the performance (pds and pfs with nearest neighbour classifier -NNC) of MARS and compares it directly to applying NNC to all prototypes (i.e. the training set). Finally, the conclusions for this work are presented.

2 MARS Design and Operation

MARS which stands for *MESO and Reduction Solution*, is a two (2) step procedure for first generating initial-prototypes, then removing those conflicting prototypes present in a local neighbourhood. MESO uses a "...data clustering approach which creates small clusters of patterns called sensitivity spheres" [KM07], while our reduction strategy adopts the nearest unlike neighbour (NUN) concept used by Dasarathy [Das94] to remove conflicting prototypes. The following sections show how MESO is used to create the initial prototypes for each class in the training set and also how NUN is used to reduce these initial prototypes.

2.1 Creating Initial Prototypes with MESO

Figure 1, displays the algorithm for creating sensitivity spheres (clusters) in MESO. As described in [KM07], the algorithm takes each instance and its closest sphere mean vector is located. If the distance between them is less than or equal to δ then the instance is added to the sphere and the sphere mean is recalculated. If however the distance between the instance and the sphere is greater than δ then δ is grown using the growth function [KM07] and the instance forms a new sphere with itself as the initial sphere mean vector. Indept details of this algorithm is not presented in this work, however the reader can further investigate the Kasten's algorithm in his paper, "MESO: Supporting Online Decision Making in Autonomic Computing Systems" [KM07].

```
Initialize the $\delta = 0$, $\mu = x_1, s_1$
For each $x_j$ sample do
    Find the nearest $\mu$ for $x_j$
        if distance from $\mu$ to $x_j \leq \delta$
            (i) add $x_j$ to $s_i$
            (ii) recompute $\mu$ using samples in $s_i$
        else
            (i) let $\delta$ = grows
            (ii) create new $s_i + 1$
            (iii) add $x_j$ to $s_i + 1$
            (iv) let $\mu = x_j$
```

Figure 1: Sphere Creation Algorithm in MESO (adapted from [KM07])

The main advantages of using MESO [KM07] to generate our initial prototypes are (1. the user does not need to decided on the number of initial prototypes to generate, the algorithm does this automatically and (2. the distance value which determines the size of a sphere is realized from the training set. With the task of choosing values for these variables taken care of by MESO, we focus on generating prototype for each class in our training set. First, the instances in the data set are grouped by their class label. Sensitivity spheres are then created for each of these groups. Once this step is completed, three instances from each sphere are selected as initial prototypes. The criteria for selecting the prototypes is as follows:

- For each sphere the instance closest to the centroid is chosen.
- The other two (2) instances are chosen as the furthest points away from the centroid of a sphere. This is done by first finding the instance, ‘ia’, furthest away from the centroid, then finding the instance furthest from ‘ia’.

Recall that one disadvantage of using a NNC is that if the classes in a data set are non-separable or overlapping, training samples in a local neighbourhood may come from different classes. As a result, test data may be misclassified using a NNC. Further, since the spheres created here by MESO is done on a class by class basis, the other classes do not have a say in the positioning of the resulting spheres. It is for these reasons that the *reduction* step of MARS is performed.

2.2 Reduction by Removing Confusing Prototypes

The purpose of removing confusing prototypes is to reduce the misclassification rate by an NNC. Figure 2 clearly features the basic algorithm used to accomplish this. Figure 3 shows the general prototype reduction at each level of a data set. A few data sets such as Lymph and Audio show a very small reduction from the original number of prototypes while others such as Breast Cancer, Vote and Tic-Tac-Toe show a radical reduction to less than 10% of the original data set. Please note that since MESO is order dependent, the reduction rate differs once the data set is shuffled.

```
1. Choose initial prototypes for each class using SSC from MESO
2. Take each prototype and find the distance (d1) of its NUN,
3. Then find the distance (d2) of its nearest like neighbour (NLN).
        If $d1 \leq d2$ , remove the prototypes
        If $d1 > d2$ , the prototype is approved and kept
4. Repeat 2 and 3 until no more prototypes can be removed
```

Figure 2: Reduction with NUN Algorithm

3 MARS Assessment

In all the literature reviewed for this work, researchers measure the performance of their PLS using classification accuracy as follows:

Data Set	All prototypes	Initial-Prototypes	Final Prototypes
Lymph	147	33	14
Iris	150	63	57
Breast Cancer	286	24	11
Heart	297	84	66
Cars	1728	69	58
Vote	430	18	8
Diabetes	768	129	92
Tic-Tac-Toe	958	21	4
Sonar	208	102	75
Balance Scale	226	21	8

Figure 3: Reduction Level of Data Sets with MARS

$$\frac{SumofCorrectClassifications}{TotalNumberofInstancesTested}$$

However, the results of this method can only be trusted when the class distribution of a data set occur with similar frequencies. In light of this, the performance of both the nearest neighbour classifier (NNC) and MARS was assessed by calculating the probability of detection (pd) and probability of false alarm (pf) [MGF07] measures for each data set used in this work. By allowing A, B, C and D to represent true negatives, false negatives, false positives and true positives respectfully, it then follows that pd also known as *recall*, is the result of true positives divided by the sum of false negative and true positives $D / (B + D)$. While pf is the result of: $C / (A + C)$.

In this section the pd and pf values of MARS and NNC are generated for ten (10) standard discrete data sets (See Figure 4) from the UC Irvine machine learning repository [AN07]. This is done by following the procedure for cross-validation experiments is described in the following section. Next our results are visualized with quartile charts Figure 6. These charts offer a non-parametric view of the result with no assumptions on the underlying distribution. To further ensure the soundness of our experiments, the Mann-Whitney U test [MW47] (a non-parametric test) was applied to our results to see if there was any statistical difference between them.

Data Set	Attributes	Instances	Labels
Lymph	18	147	4
Iris	4	150	3
Breast Cancer	9	286	2
Heart	13	297	5
Cars	6	1728	6
Vote	16	430	2
Diabetes	8	768	2
Tic-Tac-Toe	9	958	2
Sonar	60	208	2
Balance Scale	70	226	8

Figure 4: Data Set Characteristics

3.1 Experimental Method

This study used 10 x 10 way cross-validation experiments to evaluate the performance of MARS. The procedure was executed as follows:

- Shuffle the training data then divide it into N buckets.
- For each bucket, MARS is trained on nine (9) of the buckets and tested on the remaining bucket.
- The A, B, C, D values for each class are generated from the testing phase and the pds and pfs are calculated.
- The above is repeated ten (10) times collecting all pd and pf values each time.

In the end MARS is trained and tested 100 times.

3.2 Results

Figure 6 presents the quartile charts of pd and pf for KNN vs MARS over ten (10) discrete data sets. The pd and pf values are found for each class in the data sets. Although some of the have more than two(2) classes, Figure 6 presents the results of two(2) classes for each data set in column 1 and 2. For the data sets where the number of classes is greater than 2, the majority classes are chosen to be displayed. Figure 5 indicates the classes represented in each column for each data set. The results of performing the Mann-Whitney U test is indicated as (w - win, l -loss and t - tie) in parentheses next to the treatment name. A *w* or *l* signals whether or not the use of MARS is statistically different while *t* shows no difference.

Data Set	Column 1	Column 2
Lymph	fibrosis	metastases
Iris	versicolor	virginica
Breast Cancer	recurrence-events	no-recurrence-events
Heart	$> 50-1$	$> 50-4$
Cars	unacc	acc
Vote	republican	democrat
Diabetes	tested-positive	tested-negative
Tic-Tac-Toe	positive	negative
Sonar	Rock	Mine
Balance Scale	R	L

Figure 5: Data Set Classes Used

In previous work done on PLS, the goal has been that the accuracy with the use of a prototype learner is either comparable or greater than that using the entire training data. However, in this work we seek comparable or higher pd values in conjunction with lower pf values for MARS. Most of our results demonstrates that although the goal of higher pds are achieved for some data sets (lymph, diabetes, vote, breast cancer, iris, sonar), it comes at the cost of higher pfs. For instance, the median pfs for Diabetes is 50 for KNN while MARS is 100. In the cases where the median pfs for MARS are comparable or lower than those of KNN, MARS pds are lower. With these results, the use of MARS as a prototype learner must be determined by the data set and whether or not the user, in their domain, will accept a lower pd in favor of a reduced pf.

4 Conclusions and Future Work

MARS has been presented in this work. It is a prototype learner which first creates initial prototypes that are then reduced in numbers by a novel reduction method. When compared against a NNC, the results over ten (10) standard discrete data sets indicate that the use of MARS is dependent on the data set used and the parameters of acceptable results of a user and their domain.

Lymph	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (w)	50	71	100	
	KNN (l)	50	60	75	
pf	MARS (w)	11	36	57	
	KNN (l)	0	9	17	
					0 50 100

Lymph	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	38	63	89	
	KNN (w)	89	100	100	
pf	MARS (l)	0	25	44	
	KNN (w)	25	38	50	
					0 50 100

Vote	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (t)	33	80	100	
	KNN (t)	75	81	86	
pf	MARS (w)	8	38	100	
	KNN (l)	14	20	27	
					0 50 100

Vote	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	0	53	92	
	KNN (w)	72	80	86	
pf	MARS (t)	0	18	63	
	KNN (t)	14	19	25	
					0 50 100

Heart	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	13	80	100	
	KNN (w)	100	100	100	
pf	MARS (t)	0	0	0	
	KNN (t)	0	0	0	
					0 50 100

Heart	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	0	67	100	
	KNN (w)	0	100	100	
pf	MARS (t)	0	0	0	
	KNN (t)	0	0	0	
					0 50 100

Breast Cancer	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	36	91	100	
	KNN (w)	91	95	100	
pf	MARS (t)	17	86	100	
	KNN (t)	67	73	80	
					0 50 100

Breast Cancer	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (t)	0	14	83	
	KNN (t)	18	25	33	
pf	MARS (w)	0	6	61	
	KNN (l)	0	5	10	
					0 50 100

Cars	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (t)	15	67	94	
	KNN (t)	49	55	63	
pf	MARS (w)	17	37	62	
	KNN (l)	5	7	9	
					0 50 100

Cars	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (w)	0	20	57	
	KNN (l)	0	0	0	
pf	MARS (w)	4	25	67	
	KNN (l)	0	0	0	
					0 50 100

Tic-Tac-Toe	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (w)	0	72	100	
	KNN (l)	0	14	19	
pf	MARS (w)	0	77	100	
	KNN (l)	0	5	9	
					0 50 100

Tic-Tac-Toe	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	0	21	100	
	KNN (w)	0	92	100	
pf	MARS (l)	0	28	100	
	KNN (w)	80	86	100	
					0 50 100

Iris	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (t)	100	100	100	
	KNN (t)	86	100	100	
pf	MARS (w)	0	11	25	
	KNN (l)	0	0	0	
					0 50 100

Iris	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	50	80	100	
	KNN (w)	100	100	100	
pf	MARS (t)	0	0	8	
	KNN (t)	0	0	0	
					0 50 100

Sonar	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (t)	36	73	100	
	KNN (t)	67	75	85	
pf	MARS (w)	20	58	100	
	KNN (l)	30	46	55	
					0 50 100

Sonar	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	0	40	78	
	KNN (w)	44	54	67	
pf	MARS (t)	0	25	64	
	KNN (t)	15	25	33	
					0 50 100

Diabetes	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (w)	2	100	100	
	KNN (l)	86	89	91	
pf	MARS (w)	0	100	100	
	KNN (l)	45	50	58	
					0 50 100

Diabetes	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	0	0	100	
	KNN (w)	41	50	55	
pf	MARS (l)	0	0	98.2	
	KNN (w)	9	12	14	
					0 50 100

Balance Scale	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	0	19	36	
	KNN (w)	53	59	69	
pf	MARS (l)	0	8	24	
	KNN (w)	18	24	34	
					0 50 100

Balance Scale	Treatment	25%	50%	75%	Q1 median Q3
pd	MARS (l)	10	33	58	
	KNN (w)	70	77	85	
pf	MARS (t)	13	44	67	
	KNN t)	32	43	49	
					0 50 100

Figure 6: Probability of Detection (PD) and Probability of False Alarm (PF)results

References

[AN07] A. Asuncion and D.J. Newman. UCI machine learning repository, 2007.

[CH67] T. Cover and P. Hart. Nearest neighbor pattern classification. *Information Theory, IEEE Transactions on*, 13(1):21–27, Jan 1967.

[Cha74] Chin-Liang Chang. Finding prototypes for nearest neighbor classifiers. *Computers, IEEE Transactions on*, C-23(11):1179–1184, Nov. 1974.

[Das94] B.V. Dasarathy. Minimal consistent set (mcs) identification for optimal nearest neighbor decision systems design. *Systems, Man and Cybernetics, IEEE Transactions on*, 24(3):511–517, Mar 1994.

[DM02] V. Susheela Devi and M. Narasimha Murty. An incremental prototype set building technique. *Pattern Recognition*, 35(2):505 – 513, 2002.

[Har68] P. Hart. The condensed nearest neighbor rule (corresp.). *Information Theory, IEEE Transactions on*, 14(3):515 – 516, may 1968.

[KM07] E.P. Kasten and P.K. McKinley. Meso: Supporting online decision making in autonomic computing systems. *Knowledge and Data Engineering, IEEE Transactions on*, 19(4):485–499, April 2007.

[Koh90] T. Kohonen. Improved versions of learning vector quantization. pages 545 –550 vol.1, jun 1990.

[KS98] Teuvo Kohonen and Panu Somervuo. Self-organizing maps of symbol strings. *Neurocomputing*, 21(1-3):19 – 30, 1998.

[MGF07] Tim Menzies, Jeremy Greenwald, and Art Frank. Data mining static code attributes to learn defect predictors. *IEEE Transactions on Software Engineering*, 33:2–13, 2007.

[MW47] HB Mann and DR Whitney. On a test of whether one of two random variables is stochastically larger than the other. *The Annals of Mathematical Statistics*, 18, 1947.