A Support Vector Machine Based Approach to Evaluation of the Quality of Patterns

Luis Horna[1], Ricardo Barrón[2], and Salvador Godoy[3]

Centro de Inevstigación en Computación , Instituto Politécnico Nacional,
Laboratorio de Inteligencia Artificial,
Av. Juan de Dios Bátiz s/n, México, D.F., 07738, Mexico
[1]chornab08@sagitario.ipn.mx
[2]rbarron@cic.ipn.mx
[3]sgodoyc@cic.ipn.mx

Abstract. In the routine task of Pattern Recognition, time and effort is invested to extract characteristics, and create large pattern data sets that do not always behave as expected when training a pattern recognition system. In those cases it is worth to try to evaluate if the patterns that have been extracted are either of high or low quality. In this paper, we propose a method to evaluate the quality of patterns by using support vector machines.

Key words: Support Vector Machine, Quality of Patterns.

1 Introduction

One of the problems that is always present in pattern recognition is to know whether patterns from different classes are linearly separable, which can be seen as a measure of quality of the features being used. This problem can be evaluated by making scatter plots and visually evaluate, computing correlation[10] or distance between classes [4], evaluating the taxonomy of the classes, as well as making a principal component analysis (PCA) [3], [11].

When dealing with pattern recognition tasks it is common to find classes or clusters with very little separation, intersection near their borders or, in the worst case, completely overlapping Fig.(1). In such situations, it is desirable to identify which classes have this problems as well as which patterns are causing this problem. Evaluating the quality of the patterns being used is essential in order to select a classifier suitable for a given situation, if the quality of data is poor one could end up using classifiers whose computational cost is high so as to compensate. Measuring the quality of data can also be helpful to determine whether to change the features being used.

© G. Sidorov et al. (Eds.)
Advances in Soft Computing Algorithms
Research in Computing Science 49, 2010, pp. 85-91

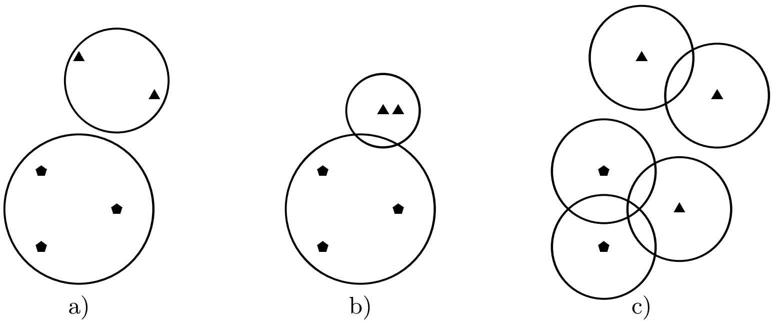


Fig. 1: a) Classes or clusters with very little separation, b) Intersection near their borders, c) Classes completely overlapping.

In this paper we propose a novel index to assess the quality of patterns by identifying three different situations:

1.- ratio of patterns from each class that could be correctly separated
2.- ratio of patterns from each class that could not be correctly separated.
3.- ratio of patterns that are classified as elements of other classes.

desirable characteristics of high-quality patterns would be to minimise this three different possibilities. In this paper we will develop a novel methodology to evaluate this three possibilities, our methodology will make use of SVM for such purpose.

2 Proposed Methodology

Assessing the quality of patterns in large data-sets is, generally, not an easy task. We have selected support vector machines (SVMs) over other classifiers, because their capability of handling large data sets [2], [7], [9],[12], as well as handling vectors of large numbers of features [8] , SVMs can deal with non-linearly separable classes , and SVMs are widely used for data mining [5], [13] which is similar to our problem.

From a formal point of view a perfectly linearly separable set of classes would be one that satisfies:

$$C_1 \cap C_2 \cap ... \cap C_n = \emptyset \tag{1}$$

where C_i is a class, now in reality

$$C_1 \cap C_2 \cap ... \cap C_n = W \tag{2}$$

the cardinality of W depends on several factors, but in general the better the quality of patterns and classifier the smallest W is.

The proposed solution in this paper makes the assumption that if a pattern is not classified correctly during training then it must be removed, under common training approaches this would be a mistake. However, since the goal in this particular problem is precisely trying to find those patterns and in consequence the classes that have problems of linear separability is ideal to remove them.

Our approach consists in two phases. First, for a fundamental set F we create a support vector machine (we use a C-SVC) for each of the C_i classes (i.e dividing F into C_i and non-C_i), then evaluate its performance and remove those patterns that were not correctly classified until performance is satisfactory, create Q_i with the reduced set C_i. Second, create a set S_i with all the patterns that could not be correctly classified when creating Q_i.

It is clear that $Q_i \cap S_i$ is not necessarily $\emptyset$, this follows from the fact that some $X_k \subset C_i$ may have ended up in S_i. It should also be pointed out that this process must have a finishing condition other than reaching some performance, because there is no warranty it will ever be achieved, instead a maximum number of iterations must be used. Once all Q_i and S_i have been computed, all that rests it to evaluate the quality of the patterns. It should be pointed out that the SVMs used have to use the same kernel function, no particular restriction is imposed when selecting the kernel function.

In order to measure the quality of patterns, we measure:

1.- ratio of patterns from each class that could be correctly separated, denoted by ψ.
2.- ratio of patterns from each class that could not be correctly separated,denoted by χ
3.- ratio of patterns that are classified as elements of other classes, denoted by φ.

each of the ratios is respectively computed by:

$$\psi_i = \frac{card(Q_i)}{card(C_i)} \tag{3}$$

where values close to zero represent very low pattern quality, and values close to one represent high quality patterns.

$$\chi_i = \frac{card(\overline{Q_i \cap C_i})}{card(C_i)} \tag{4}$$

χ_i represents exactly the opposite of ψ_i.

$$\varphi_i = \frac{card(S_i)}{card(F)} \tag{5}$$

finally, if φ_i is very close to zero it means that very few patterns are causing problems when classifying C_i, and values very close to one are a sign that almost every pattern causes problems.

After computing ψ_i, χ_i, and φ_i, it is clear that Q_i and S_i contain only the patterns that could be classified with the highest accuracy, this could be used to assign weights to each class C_i, and Q_i could be used as a model to create auto-associative memories that work as a filter.

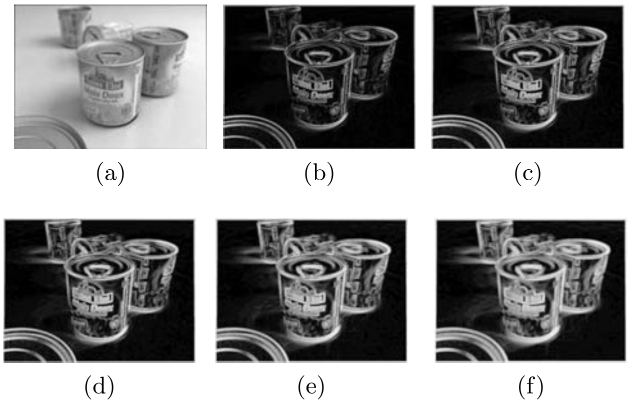
At this point two situations must be clarified, first our approach uses the SVMs to detect which patterns are causing problems when training, on eahch iteration we only keep those that are useful. Second, the proposed indexes aim to measure the quality of patterns of a given data set taking into account the patterns that cause problems during training.

3 Experimental Results

In this section, we present two experiments, one to evaluate poor quality patterns another with high-quality. The first experiment was an attempt to create pattern recognition system that could establish relationship between an image I and a depth map D. The second experiment consist in evaluating the quality of well known data sets.

In the first experiment, the features extracted are the standard deviations of the image using different scales of 3x3,5x5,7x7,9x9, and 11x11, which are used to create a feature vector of dimension 5. Image I is sampled only along the edges, and the fundamental set has the form:

$F = \{((L_{3x3}, L_{5x5}, L_{7x7}, L_{9x9}, L_{11x11}), d_1), ...((L_{3x3}, L_{5x5}, L_{7x7}, L_{9x9}, L_{11x11})), d_n)$


(a) (b) (c)

(d) (e) (f)

Fig. 2: Images representing the characteristic extracted.

where $d_i \in [0, 255]$, Fig.(2) shows an image and each of the features measured. In order to make this experiment easy to understand, depths are grouped in intervals of five, for instance C_1 would group all d_i in $[0 - 4]$. The SVMs use a radial basis kernel with parameters $\gamma = 0.20$ and $C = 1$.

From an empirical point of view it may seem that the features from Fig.(3) should provide information to accomplish the goal. However, when applying

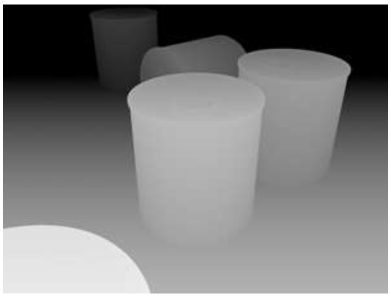


Fig. 3: Depth map used as target.

our methodology the following information about the patterns being used is obtained, tab.(1) shows the results for five classes.

Table 1: Result of the first experiment.

class	ψ	χ	φ
0-4	0.72	0.28	0.01
89-93	0.0	1.0	0.17
224-228	0.16	0.84	0.04

results show that the features are of poor quality and further more classes are completely mixed, which makes data completely useless. This experience shows that it is worth to evaluate the quality of patterns before investing time and effort training and testing different classifiers. The second experiment consists in evaluating the quality of patterns from the well known Iris plant (three classes) and Wine (three classes)[18] data sets, results of their evaluations are shown in tab.(2), tab.(3) and tab.(4), parameters used in the SVM are $\gamma = 1/N$ and $C = 1$,with N =dimension of feature vector.

Table 2: ψ values

Data set	ψ_1	ψ_2	ψ_3	ψ_4
Iris plant	1	0.91	1.0	n/a
Wine	0.93	0.97	1.0	n/a

Table 3: χ values

Data set	χ_1	χ_2	χ_3	χ_4
Iris plant	0.0	0.09	0	n/a
Wine	0.07	0.03	0.0	n/a

Table 4: φ values

Data set	φ_1	φ_2	φ_3	φ_4
Iris plant	0.0	0.05	0.05	n/a
Wine	0.02	0.01	0.01	n/a

From these results it can be confirmed that the quality of this well known training sets is good, note that the classes that have small value of ψ_i are those classes that are mixed in some degree with other classes. Which is expected as is commonly reported in literature.

In this section we have presented how our methodology works on both high-quality patterns and completely useless patterns, we have also shown the capability of our approach to detect classes that are overlapped, in all experiments libsvm [1] was used.

4 Future Work

It remains to evaluate our method on other patterns recognition approaches such as neural networks or distance based classifiers. Another interesting aspect to develop in future research is the computing of weights for classes using our methodology. It also would be interesting to investigate the use of Q_i to create auto-associative memories, which could be used to improve the accuracy of classification when using a given classifier. Finally it remains to expand our methodology to detect relevant/irrelevant features from patterns.

5 Conclusion

In this paper a novel method for evaluating the quality of patterns has been presented, also it has been shown how our method can be used to create sets of "model" patterns, which could be later used to filter unknown patterns. We believe that the proposed methodology could be an alternative to exiting techniques for evaluating the quality of patterns. From our perspective the proposed methodology is useful because it extracts the set of patterns causes most of the problems when training, which could be analized to either change the features used or label such set as new class.

References

1. C-C.Chang, C-J.Lin: LIBSVM: a library for support vector machines. http://www.csie.ntu.edu.tw/ cjlin/libsvm/ (2001)
2. Y.L. Murphey, Chen Zhihang, M. Putrus, L. Feldkamp: SVM learning from large training data set. In: Proceedings of the International Joint Conference on Neural Networks, Vol.4, pp. 2860 - 2865 (2003)
3. A.V. Anghelescu, I.B. Muchnik: Combinatorial PCA and SVM methods for feature selection in learning classifications (applications to text categorisation), In: KIMAS '03, pp. 491-496 (2003)
4. I. W. Tsang, J. T. Kwok, P.-M. Cheung: Core vector machines: fast SVM training on very large data sets.Journal of Machine Learning Research, 6, pp. 363-392 (2005)
5. L. Yu, S. Wang, K. K. Lai: Mining Stock Market Tendency Using GA-Based Support Vector Machines, Internet and Network Economics. Vol. 3828/2005, Berlin, Springer, pp. 336-345 (2005)
6. J. Cervantes, X. Li, W. Yu , J. Bejarano: Multi-Class Support Vector Machines for Large Data Sets via Minimum Enclosing Ball Clustering. In : CEEE 2007, Mexico (2007)

7. J. Cervantes, X. Li, W. Yu , J. Bejarano: Two-stage svm classification for large data sets via randomly reducing and recovering training data. In: IEEE International Conference on Systems, Man and Cybernetics, ISIC., pp. 3633 - 3638 , Montreal, Canada (2007)
8. Henryk Maciejewski: Quality of Feature Selection Based on Microarray Gene Expression Data. Computational Science – ICCS 2008, Vol. 5103/2008, pp. 140-147, Berlin, Springer (2008)
9. Zhi-Qiang Zeng, Hua-Rong Xu, Yan-Qi Xie, Ji Gao: A geometric approach to train SVM on very large data sets. In: ISKE 2008., Vol.1, pp. 991 - 996, Xiamen, China (2008)
10. Jirong Li: Feature Selection Based on Correlation between Fuzzy Features and Optimal Fuzzy-Valued Feature Subset Selection, In: IIHMSP '08, pp. 775 - 778, Harbin, China (2008)
11. Yihui Luo, Shuchu Xiong, Sichun Wang: A PCA Based Unsupervised Feature Selection Algorithm. In: WGEC '08, pp. 299 - 302, Hubei, China (2008)
12. J. D. Wang, H. C. Liu: Evaluating the Ambiguities between Two Classes via Euclidean Distance. Asian Journal of Health and Information Sciences, Vol. 4, No. 1, Taiwan, pp. 21-35, (2009)
13. P.C, C.Y. Tsai, C.H Huang, C.Y. Fan: Application of a Case Base Reasoning Based Support Vector Machine for Financial Time Series Data Forecasting, Emerging Intelligent Computing Technology and Applications, Vol. 5755/2009 ,pp. 294-304, Berlin, Springer (2009)
14. E.Y. Cheu, C. K. Kwoh, Z. Zhou: On the Two-level Hybrid Clustering Algorithm, Nanyang Technological University, Singapore.
15. C. van der Walt, E. Barnard: Data characteristics that determine classifier performance. Human Language Technologies Research Group Meraka Institute, Pretoria: South Africa.
16. C. Domeniconi, D. Gunopulos: Efficient Local Flexible Nearest Neighbor Classification, Dept. of Computer Science, University of California, Riverside: USA.
17. L. Zhang, F. Lin, B. Zhang: Support vector machine learning for image retrival. State Key Laboratory of Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University, Beijing: China.
18. UCI machine learning repository. http://archive.ics.uci.edu/ml/.