

Clustering Inside Classes Improves Performance of Linear Classifiers

Dmitriy Fradkin
Siemens Corporate Research
755 College Rd. East
Princeton, NJ 08540
dmitriy.fradkin@siemens.com

Abstract

This work systematically examines a Clustering Inside Classes (CIC) approach to classification. In CIC, each class is partitioned into subclasses based on cluster analysis. We find that CIC, by extracting local structure and producing compact subclasses, can improve performance of linear classifiers such as SVM and logistic regression. It is compared against a global classifier on four benchmark datasets. We empirically analyze effects of the training set size and the number of clusters per class on the results of the CIC approach. We also examine use of an automated method for selecting the number of clusters for each class.

1 Introduction

In this work we analyze Clustering Inside Classes (CIC) approach, previously described in [7, 5], that combines class labels with cluster analysis as part of building a classifier. We show that this approach can lead to improved accuracy over the state of the art linear classifiers, such as regularized logistic regression [13, 10] and Support Vector Machines (SVM) [14]. We evaluate the role of dataset characteristics (such as the size of the training set) in obtaining such improvements, as well as the effect of parameters such as number of clusters per class. These results are supported by extensive experimentation on the benchmark datasets from the UCI repository [1].

In Section 2 we describe the algorithm. In Section 3 we describe experimental work. In Section 4 we contrast our work and prior research. In Section 5 we summarize the main contributions of our work.

A detailed discussion of related work and full experimental analysis are presented in the longer online version of this paper¹.

¹<http://paul.rutgers.edu/~dfradkin/papers/cic-ictai08-long.pdf>

2 CIC Approach

The CIC classification scheme is described in Algorithm 1. During the training stage each class is partitioned into k clusters (lines 1-3), and then classifier R is trained to classify a new point into one of the resulting clusters. At classification time, a new point is assigned to some cluster i (line 1 or classification stage), which is then mapped a unique class label that is returned as prediction (line 2 of the classification stage). When $k = 1$, the algorithm produces a regular K -class classifier.

Note that the scheme described above partitions each class into the same number of clusters, k . Our experimental results will show that even under such a restriction the CIC approach can improve performance of state of the art linear classifiers. However, this restriction is clearly not necessary, since k could be set separately for each class, by the user or automatically. A number of heuristics exist for automatically choosing the number of clusters [11, 3] when applying a clustering algorithm, however there is no single standard approach. We will experiment with automatically selecting the number of clusters in each class using the method of [12] and show that it performs better than the global classifier.

Previous work on synthetic data [5] suggest that CIC may be particularly useful when a class consists of loosely disconnected components or has an "odd" shape. Even when we cannot be certain that these conditions hold (for example due to high dimensionality of the real world data), the CIC approach can improve performance. It is able to do so by partitioning classes into convex subclasses that can be easily separated by linear classifiers. This has the additional benefit of identifying class structure of the data [5].

3 Experimental Validation

3.1 Methods and Datasets

K-Means [4, 9] is used to find clusters in each class. We experiment with a set of different pre-specified values for

k	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 8$	$k = 10$	$k = 15$	X
image	92.10	91.71	89.38 ³	89.90 ³	89.71 ³	89.57 ³	90.19 ²	89.81 ³	89.67 ³	89.91 ³
pendigit	95.85	97.48 ³	97.51 ³	98.00³	97.71 ³	97.74 ³	97.88 ³	97.68 ³	97.57 ³	97.97 ³
satimage	86.05	85.90	87.30	88.45 ³	89.35 ³	89.45 ³	89.75 ³	89.60 ³	90.15³	90.00 ³
vowel	51.08	53.25	56.06 ¹	56.71¹	52.16	54.11	51.52	53.68	51.30	53.25

Table 1. SVM results (% accuracy). Note that on 3 of the 4 datasets, CIC-SVM using X-Means to determine k for each class (X column) is significantly better than SVM alone.

k	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 8$	$k = 10$	$k = 15$	X
image	91.67	91.19	88.95 ³	87.33 ³	88.33 ³	88.81 ³	89.62 ²	89.90 ¹	88.95 ³	88.86 ³
pendigit	92.08	95.83 ³	95.71 ³	96.17 ³	96.23 ³	96.54 ³	96.31 ³	96.57³	96.43 ³	96.57³
satimage	83.90	84.35	85.30 ²	86.90 ³	87.90³	87.65 ³	87.45 ³	87.10 ³	87.05 ³	87.25 ³
vowel	46.54	49.13	51.52	46.10	46.32	48.05	45.24	44.59	45.89	48.05

Table 2. BMR results (% accuracy). Note that on 3 of the 4 datasets, CIC-BMR using X-Means to determine k for each class (X column) is significantly better than BMR alone.

Algorithm 1 Training and Classification with CIC

Require: A set W with $K \geq 2$ classes, an integer $k \geq 1$.
 {Training with CIC}

- 1: **for** $j = 1, \dots, K$ **do**
- 2: Partition class L_j into k clusters.
- 3: **end for**
- 4: Train classifier R using all training data to recognize all $k \cdot K$ clusters.

Require: A point x . {Classification with CIC}

- 1: Let $i = R(x)$, $i = 1, \dots, k, \dots, k \cdot K$.
 - 2: Return class of cluster i .
-

the number of clusters k in each class, as well as with selecting k automatically for each class using the X-Means software [12]. As our classifiers we use LIBSVM [2] and Bayesian Multinomial Regression (BMR) [10].

For the experiments we used four well-known datasets from UCI [1]: **Image Segmentation**, **Pendigit**, **Satellite Image (Satimage)** and **Vowel**. In all of these datasets, except for Satimage, the relative class sizes are approximately the same in the tests sets as in the training sets.

Details on parameter selection for these methods, as well as specifics on preprocessing and training/test splits of the datasets are discussed in the long version of the paper.

3.2 Clustering Inside Classes on Benchmark Data

Tables 1 and 2 shows the accuracy of CIC on each dataset for different number of clusters per class, $k = 1, 2, 3, 4, 5, 6, 8, 10, 15$ and with X-Means. Superscripts ¹,

², ³ will be used to mark results that are significantly different (at 0.95, 0.99 and 0.999 confidence levels respectively) from the performance of the basic classifier ($k = 1$), as determined using McNemar Test.

It is clear from these results that on all datasets except for Image CIC leads to improvement in classification accuracy for at least some values of k . With the SVM, the results on the Image dataset with $k > 2$ are significantly worse at 0.99 level than those of the basic classifier. On the other hand, the accuracy of CIC is significantly better at 0.999 level for all values of k (except $k = 15$) on the Pendigit dataset and for $k > 3$ on the Satimage dataset (again, except $k = 15$); and is somewhat better (0.95 level) for $k = 3, 4$ on the Vowel dataset. The results with BMR are qualitatively similar (significant improvements on the Pendigit and Satimage datasets, no significant difference on the Vowel dataset, and worse results on the Image dataset).

Notice that the value of k for which the best results are obtained appears related to the number of points per class in the training set. On the Image dataset this number is on average 30, and the best results are for $k = 1$ (basic classifier). For the other, larger, datasets the performance improves as k increases up to a point, and then starts to decrease. Intuitively, increasing the number of clusters leads to clusters with fewer point, which in turn makes it difficult to train good classifiers for distinguishing them. This argument is both intuitively accessible and can be supported by learning theory [14]. However, as shown in Section 3.4, these differences in performance are not completely due to the training set size, but are also affected by the intrinsic structure of the classes. In other words, Image dataset has classes that

consist of a single component, while the other datasets have classes with a more complicated structure.

Using X-Means does not give the best results, which is not surprising since it needs to determine k for each class, while the best result over different fixed k is selected *a posteriori*. However, X-Means leads to consistently better (significantly so on the Pendigit and Satimage datasets) results than the global classifier, except for the Image dataset. In other words, where improvement with CIC is possible, using X-Means to determine the number of clusters also leads to an improvement.

3.3 Results with Non-Linear Classifiers

Here we examine performance of CIC using SVM with RBF kernel. Results for different values of k are in Table 3.

The results of the RBF SVM alone on all datasets were higher with any linear classifier, and also somewhat better than most combinations of CIC with linear classifiers. Using CIC (whether with a fixed k or with X-Means) with RBF SVM did not result in any improvements for Image and Vowel datasets, and there was only minor (not statistically significant) improvement on Pendigit and Satimage datasets for intermediate values of k , but not for X-Means. Clearly, non-linear methods such as RBF SVM can represent more complicated decision boundaries - and separate odd-shaped classes better - and therefore stand to benefit less from CIC than linear methods.

These observations support the intuition that CIC improves the performance of linear classifiers by increasing the number of linear decision surfaces and, in effect, approximating with them the non-linear separation boundaries between different classes.

3.4 The Effect of the Training Set Size

We conducted experiments on the benchmark datasets to examine the role of the training set size, both in absolute terms and as a proportion of the data. This was achieved by varying the fraction of the points in each dataset that was used for training and by varying the number of clusters k . Details are given in the longer paper.

In short, the experiments showed that the use of CIC did not lead to worse results when sufficient training data was present. On Pendigit and Satimage datasets CIC improves results for almost setting of parameters. On the Image dataset CIC does not decrease accuracy when the training set was sufficiently large (50% of the data or 175 points per class in the training set). Failure of CIC to produce improvements on Image suggests that classes there have a single intrinsic component.

4 Discussion

While the CIC approach seems simple, it was first discussed, to the best of our knowledge, only in [7]. Since then others [15, 8] have investigated different aspects of such approach, but left gaps that our work aims to fill.

Unlike [7] we obtain a single good clustering result and then build a single multiclass classifier. Therefore, any improvements observed are exclusively due to CIC, and not to bagging or boosting of individual CIC classifiers.

[7] also did not address the choice of k , leaving it up to the user. Our results on multiclass problems (as opposed to the two-class problems discussed in [7]), using state-of-the-art classifiers, show that CIC approach improves performance of the linear classifiers. Unlike [15], we examine applicability of the CIC approach to different linear classifiers, not only Naive Bayes, and to non-linear classifiers. We also experiment with an efficient method for automatically selecting k for each class [12], instead of using computationally expensive cross-validation.

To our knowledge ours is the first work to extensively examine the CIC approach on its own in a general setting (i.e. not tied specifically to problems with class imbalances, or to a particular classifier method or a combination of classifiers). We consider in much greater detail than previous work the effects of the number of clusters, k , and of the training set and class size on the performance of the CIC approach. We also experiment with an *efficient* method for automatically selecting k for each class - an important issue that was not previously addressed.

5 Summary

We have shown that Clustering Inside Classes (CIC) improves accuracy of linear classifiers provided that sufficient training data is available and that the classes have more than one intrinsic component. Even when classes have only one such component, CIC does not cause performance to deteriorate provided there is enough training data for each class.

We examined CIC with different training set and class sizes, using both user-specified and automatically determined values for the number of cluster in a class, k . The choice of k does affect the results, though there tend to be whole ranges of values of k for which CIC improves performance of linear classifiers. We have observed improvements with $10 \geq k \geq 3$ for all but the Image dataset. Using the approach of [12] for automated selection of k improved results compared to a single global classifier. These results were only slightly worse those with the best values of k selected for each dataset *a posteriori*. Prior works did not propose an efficient method for automatic selection of k .

The accuracy improvements obtained with CIC and linear classifiers do not match results of a non-linear classifier.

k	$k = 1$	$k = 2$	$k = 3$	$k = 4$	$k = 5$	$k = 6$	$k = 8$	$k = 10$	$k = 15$	X
image	92.71	91.14 ³	90.38 ³	90.67 ³	90.33 ³	90.19 ³	90.19 ³	90.00 ³	89.48 ³	90.42 ³
pendigit	97.80	97.68	97.74	97.74	97.94	97.71	97.88	97.68	97.48	97.77
satimage	91.60	91.55	91.05	90.55 ¹	91.60	91.70	91.65	90.95	90.30	90.65
vowel	66.45	59.31 ³	56.49 ³	56.49 ³	54.98 ³	54.55 ³	51.73 ³	52.38 ³	50.87 ³	51.73 ³

Table 3. RBF SVM results (% accuracy). Using X-Means does not lead to improvements. In general CIC with RBF SVM does not improve on RBF SVM alone.

Also, CIC fails to improve performance of non-linear classifier. However, CIC allows for other advantages:

- Linear classifiers tend to be faster to train than non-linear ones. Training several independent linear SVM (which is also easily parallelizable) is thus more efficient than building a non-linear SVM.
- Linear classifiers tend to be faster at classification time. In order to apply a linear SVM to a new case only one similarity (inner product) computation is needed. Therefore, only k such computations are needed to apply CIC. However, applying non-linear SVM requires computing similarity with each support vector, of which there are frequently hundreds or thousands.
- Linear classifiers are much more interpretable. Their weights indicate which features are important for classification and how they affect predictions. In many applications being able to understand how classifier makes predictions is almost just as important as having a high accuracy.

An interesting avenue for further work is to combine CIC with non-linear classifiers by performing cluster analysis in a non-linear space. A number of techniques for performing such analysis have been described, for example in [6].

The datasets examined in our work arose in signal-processing applications. Text and bioinformatics datasets are frequently characterized by large number of features, sometimes exceeding the number of available training points. While we believe that the CIC approach will also be successful in such applications, the characteristics of the data would have to be taken into account, most likely by utilizing a more appropriate clustering approach.

Another direction for future investigations is development of visualization/representation methods for the extracted class structure (i.e. within-class clusters) as a data exploration method.

The results described here demonstrate usefulness of and provide insights into the CIC approach and encourage further investigations and applications.

References

- [1] C. Blake and C. Merz. UCI repository of machine learning databases, 1998.
- [2] C.-C. Chang and C.-J. Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at <http://www.csie.ntu.edu.tw/~cjlin/libsvm>.
- [3] C. Farley and A. E. Raftery. Model-based clustering, discriminant analysis and density estimation. *J. of the American Statistical Association*, 97(458):611–631, June 2002.
- [4] E. Forgy. Cluster analysis of multivariate data: efficiency vs interpretability of classification. *Biometrics*, 21:768, 1965.
- [5] D. Fradkin. *Within-Class and Unsupervised Clustering Improve Accuracy and Extract Local Structure for Supervised Classification*. PhD thesis, Rutgers, The State University of New Jersey, January 2006.
- [6] M. Girolami. Mercer kernel based clustering in feature space. *IEEE Transactions on Neural Networks*, 13(3):780–784, 2002.
- [7] N. Japkowicz. Supervised learning with unsupervised output separation. In *Proceedings of the IASTED ASC'2002*, pages 321–325, 2002.
- [8] P. W. Junjie Wu, Hui Xiong and J. Chen. Local decomposition for rare class analysis. In *Proceedings of the KDD 2007*, pages 814–823, 2007.
- [9] S. Lloyd. Least squares quantization in PCM. *IEEE Transactions on Information Theory*, 28:128–137, 1982.
- [10] D. Madigan, A. Genkin, D. D. Lewis, S. Argamon, D. Fradkin, and L. Ye. Author identification on the large scale. In *Proceedings of Joint Annual Meeting of the Classification Society of North America*, 2005. Software: www.bayesianregression.org/bmr.html.
- [11] G. W. Milligan and M. C. Cooper. An examination of procedures for determining the number of clusters in a data set. *Psychometrika*, 58(2):159–179, 1985.
- [12] D. Pelleg and A. Moore. X-means: Extending k-means with efficient estimation of the number of clusters. In *Proceedings of the ICML'00*, pages 727–734, 2000.
- [13] R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society. Series B (Methodological)*, 58(1):267–288, 1996.
- [14] V. Vapnik. *The Nature of Statistical Learning Theory*. Springer-Verlag, 2nd edition, 1998.
- [15] R. Vilalta and I. Rish. A decomposition of classes via clustering to explain and improve naive bayes. In *Proceedings of ECML'03*, pages 444–455, 2003.