CLOOSTING: CLustering data with bOOSTING

F. Smeraldi[1], M. Bicego[2,3], M. Cristani[2,3], and V. Murino[2,3]

[1] School of Electronic Engineering and Computer Science, Queen Mary University of London, UK
[2] Computer Science Department, University of Verona, Italy
[3] Istituto Italiano di Tecnologia (IIT), Italy

Abstract. We present a novel clustering approach, that exploits boosting as the primary means of modelling clusters. Typically, boosting is applied in a supervised classification context; here, we move in the less explored unsupervised scenario. Starting from an initial partition, clusters are iteratively re-estimated using the responses of one-vs-all boosted classifiers. Within-cluster homogeneity and separation between the clusters are obtained by a combination of three mechanisms: use of regularised Adaboost to reject outliers, use of weak learners inspired to subtractive clustering and smoothing of the decision functions with a Gaussian Kernel. Experiments on public datasets validate our proposal, in some cases improving on the state of the art.

1 Introduction

Boosting algorithms [9] are a class of ensemble methods that have repeatedly proved highly effective in the context of supervised classification. A somehow less explored scenario is the use of boosting techniques for unsupervised classification, namely clustering. Recently, there have been a few attempts to extend the boosting approach to the clustering domain. Some authors have proposed to combine different clustering algorithms in a boosting-like framework. For instance [20] and [10] introduced weighted re-sampling of data points according to how reliably they are classified. In [16], a general iterative clustering algorithm is presented that combines several algorithms by keeping track both of point weights and of a membership coefficient for each pair of a point and a model. Point weights are updated by a Bregman divergence optimisation procedure very similar to Adaboost [8].

All these works represent attempts at combining different clustering algorithms in an Adaboost-like framework. However, an alternative stance consists in using the Adaboost algorithm itself to describe the data, moving from the "boosting clustering methods" paradigm to "clustering *with* boosting methods" paradigm. This is in line with some other recent and very successfully approaches, based on Support Vector Machines [3, 2]. The approach presented in this paper explores this direction, and is aimed at developing a clustering technique which employs a boosted classifier to describe each cluster. This may be particularly advantageous for clustering since it is known that, depending on the chosen weak learners, boosting classifiers may describe non convex non connected regions in the feature space. A somehow related approach is the very recent MCBoost [11] where a noisy-or scheme is used to combine the output of several boosted classifier, one per cluster; however the weight update equations for the single boosting algorithm depend in this case on the decision of the algorithms after fusion, so that strictly speaking the strong classifiers are not obtained by Adaboost.

In this work, we explore the use of Adaboost proper to model the clusters, via a cycle of learning and classification iterations reminiscent of k-means – in line of what has been done by Camastra and Verri with Support Vector Machines [3]. Contrary to [11] we choose to model the data with a regularised Adaboost algorithm, namely Adaboost-REG [18]. The specificity of the clustering problem is dealt with at the level of the weak learners. These are balls in feature space, centred at the locations identified by a potential similar to what is done in Subtractive Clustering [4, 5]. The locality of clusters is enforced by spatial smoothing of the output of the strong classifiers.

The suitability of the proposed approach has been tested on 4 different standard ML datasets, yielding results that compare favourably with the state of the art. Interestingly, the experiments suggest that our algorithm is especially advantageous in higher-dimensional spaces.

The rest of the paper is organised as follows: Section 2 presents an introduction to the Adaboost algorithm, followed by a discussion of related topics and variants of the algorithm that are relevant to its application to the clustering problem. Our proposal is detailed in Section 3, followed by experimental results in Section 4. A brief discussion in Section 5 concludes the paper.

2 Standard and Regularised Adaboost

Boosting concerns itself with the problem of combining several prediction rules with poor individual performance into a highly accurate predictor. This is commonly referred to as combining a set of "weak" learners into a "strong" classifier. Adaboost, introduced by Yoav Freund and Robert Schapire [8], differs from previous algorithms in that it does not require previous knowledge of the performance of the weak hypotheses, but it rather adapts accordingly. This adaptation is achieved by maintaining a distribution of weights over the elements of the training set.

Adaboost is structured as an iterative algorithm, that works by maintaining a distribution of weights over the training examples. At each iteration a new weak learner (classifier) is trained, and the distribution of weights is updated according to the examples it misclassifies; the underlying idea is to increase the importance of the training examples that are "difficult" to classify. The weights also determine the contribution of the each weak learner to the final strong classifier.

Unfortunately, as the number of iterations increases Adaboost tends to put higher weight on examples that are difficult to classify (see [18] and references

Table 1. The Adaboost and Adaboost-REG algorithms. For Adaboost, $C = 0$ and the minimisation (step 1) can be solved in closed form to give $\alpha_t = (\log(1 - \epsilon_t) - \log \epsilon_t)/2$

Adaboost and Adaboost-REG:

Initialise $d_{1,i} = \frac{1}{m}$.
For $t = 1, \ldots, T$:

1. Select the weak learner h_t that minimises the training error ϵ_t, weighted by current distribution of weights $d_{t,i}$: $\epsilon_t = \sum_{i|u_{t,i}=-1} d_{t,i}$.
2. Find the coefficient α_t that minimises
$$Z(\alpha_t, \boldsymbol{\alpha}_{t-1}) = \sum_i \exp\left(-\rho(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}_t) + C\zeta(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}_t)\right) \tag{1}$$
where $\rho(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}_t) = \sum_{\tau=1}^{t} \alpha_\tau u_{\tau,i}$ and $C\zeta(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}_t)$ is a regularization term; abort if $\alpha_t = 0$ or $\alpha_t \geq \mathit{\Gamma}$ where $\mathit{\Gamma}$ is a large constant.
3. Compute the new weights according to $d_{t,i} \propto \exp(-\rho(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}_t))$ and normalise.

Output the final classifier: $H(\boldsymbol{x}) = \text{sign}\left(F(\boldsymbol{x})\right)$, where
$$F(\boldsymbol{x}) = \sum_{t=1}^{T} \alpha_t h_t(\boldsymbol{x}) \tag{2}$$

therein). In high-noise cases these often happen to be outliers. This is of particular relevance to our clustering application because of the arbitrariness in the initial choice of the clusters (see Section 3).

A number of algorithms have been developed to alleviate this issue [7, 18, 6, 22, 21]. In the Adaboost-REG algorithm [18] this is done by substituting the *soft margin* of a point for its hard margin. This is an estimate of the hard margin corrected by a measure of how much we trust the point.

In the following, we will indicate the m training examples as $\{(\boldsymbol{x}_1, y_1), \ldots, (\boldsymbol{x}_m, y_m)\}$, where the $\boldsymbol{x}_i$ are vectors in a feature space X and the $y_i \in \{-1, +1\}$ are the respective class labels. Given a family of weak learners $\mathcal{H} = \{h_t : X \to \{-1, +1\}\}$, it is convenient to define $u_{t,i} = y_i h_t(\boldsymbol{x}_i)$, so that $u_{t,i}$ is $+1$ if h_t classifies $\boldsymbol{x}_i$ correctly, and -1 otherwise. If we now let $d_{t,i}$ be the distribution of weights at time t, with $\sum_i d_{t,i} = 1$, the cycle of iterations can be described as in Table 1 in a way that summarises both Adaboost and Adaboost-REG.

The regularization term ζ that appears in Equation 1 is used in Adaboost-REG to quantify the "mistrust" of each particular point. It is defined based on a measure of influence of the pattern on the combined hypothesis [18]:

$$\zeta(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}_t) = \left(\sum_t \alpha_t d_{t,i}\right)^2, \tag{3}$$

based on the intuition that a pattern that is often misclassified will have a high influence. Note that the minimisation in Equation 2 must be carried out numerically with some linear search algorithm, e.g. the Golden Section algorithm [17].

2.1 Using weak learners that abstain

Because of the local nature of the clustering problem, we would like to be able to use weak learners that specialise on a particular region of feature space. This can be done by allowing weak learners to abstain on part of the feature vectors. To this aim, we extend to the case of Adaboost-REG one of the estimates for Alpha reported in [19], specifically:

$$\alpha = \frac{1}{2}\ln\left(\frac{W_+ + W_0/2}{W_- + W_0/2}\right). \tag{4}$$

where W_+ is the sum of the weight of the examples that the weak hypothesis classifies correctly, W_- is the misclassified weight and W_0 the weight on which the hypothesis abstains.

We extend this to Adaboost-REG by adding the following contribution to $Z_t(\alpha_t, \boldsymbol{\alpha}_{t-1})$ for each point i on which the weak learner h_t abstains:

$$\frac{1}{2}e^{-\rho_t^+(\boldsymbol{u}_{t,i},\boldsymbol{\alpha}_t)+C\zeta(\boldsymbol{u}_{t,i})} + \frac{1}{2}e^{-\rho_t^-(\boldsymbol{u}_{t,i},\boldsymbol{\alpha}_t)+C\zeta(\boldsymbol{u}_{t,i})}. \tag{5}$$

where

$$\rho_t^+(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}) = \rho_{t-1}(\boldsymbol{u}_{t-1,i}, \boldsymbol{\alpha}_{t-1}) + \alpha_t \tag{6}$$

and

$$\rho_t^-(\boldsymbol{u}_{t,i}, \boldsymbol{\alpha}) = \rho_{t-1}(\boldsymbol{u}_{t-1,i}, \boldsymbol{\alpha}_{t-1}) - \alpha_t. \tag{7}$$

Thus the penalty term for the point is calculated by treating half of its weight as misclassified and the other half as correctly classified; this is a generalisation to the regularised case of the expression for α in Equation 4.

3 Iterative clustering with Adaboost

We propose an iterative clustering scheme outlined in Table 2. Clusters are iteratively re-estimated using one-vs-all boosted classifiers.

Within-cluster homogeneity and separation between the clusters are obtained by a combination of three mechanism:

1. use of regularised Adaboost to reject outliers, as detailed in Section 2;
2. use of weak learners (possibly localised) inspired to Subtractive Clustering, as explained in Section 3.1 below;
3. smoothing of the decision functions with a Gaussian kernel, see point 2 in Table 2.

Table 2. Iterative clustering with Adaboost

```
The Cloosting algorithm:

Input: points $\{\boldsymbol{x}_i\}$, number of clusters $K$.

Assign initial cluster labels $k_i$ to each $\boldsymbol{x}_i$ either at random or based on a k-means run

Repeat until convergence:

  1. Train $K$ strong classifiers $F_k(\boldsymbol{x})$ to recognise the elements of each cluster, using
     the Adaboost-REG algorithm in Table 1
  2. Compute the scores $S_k(\boldsymbol{x}_i) = \sum_j F_k(\boldsymbol{x}_j) \exp(-\beta \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2)$
  3. Assign each point $\boldsymbol{x}_i$ to cluster $k_i = \arg\max_k S_k(\boldsymbol{x}_i)$

Output the final clusters $\{(\boldsymbol{x}_i, k_i)\}$
```

Regularization through smoothing in a boosted clustering setting also appears in [1], where however it is applied to a separate set of 2D spatial coordinates associated to the feature vectors and not to the vectors $\boldsymbol{x}_i$ themselves. The spatial weak learners used in [1] are ineffective in our context, while smoothing the final decision functions greatly improves the stability of the algorithm.

Experimentally, the algorithm is found to reliably converge within a limited number of iterations; for details, see Section 4.

3.1 "Subtractive" weak learners

We use spherical neighbourhoods ("balls") in feature space as weak learners. A weak learner classifies points that fall inside the ball as belonging to the cluster; the other points are rejected (i.e. assigned to any other cluster).

When training the boosted classifier for cluster k, at each iteration the centre of a ball learner is chosen as the point $\boldsymbol{c}_{k,t} = \boldsymbol{x}_{i^*(t)}$ that maximises the following weighted data density:

$$D(\boldsymbol{x}_{i^*(t)}) = \sum_{j|\boldsymbol{x}_j \in k} d_{j,t} e^{-\gamma \|\boldsymbol{x}_i - \boldsymbol{x}_j\|^2} - \Delta(\boldsymbol{x}_i, \boldsymbol{d}) \tag{8}$$

where $\gamma = 4/r_a^2$, r_a is the effective radius of the neighbourhood on which the average is computed, and $\Delta(\boldsymbol{x}_i, \boldsymbol{d})$ is a penalty term that discounts previously chosen centres:

$$\Delta(\boldsymbol{x}_i, \boldsymbol{d}) = \sum_{\tau=1}^{t-1} d_{i^*(\tau),\tau} e^{-\delta \|\boldsymbol{x}_i - \boldsymbol{x}_{i^*(\tau)}\|^2}. \tag{9}$$

This is a weighted version of the density function used for the selection of cluster centres in Subtractive Clustering [4, 5]. Once a centre is chosen, the radius of the ball is optimised for the best weighted error rate, according to the Adaboost distribution of weights.

Table 3. Description of the datasets used for testing.

Dataset	n. of clusters	n. of patterns	n. of features
Iris	3	150	4
Biomed	2	194	5
WBC	2	683	9
Ionosphere	2	351	34

It is observed that the number of centres selected adaptively by Adaboost for representing each cluster often diminishes as better centres are chosen in later iterations of Cloosting.

3.2 Weak learners that abstain

In order to increase the locality of the clustering process, we tested our algorithms using weak learners that specialise on a local area in feature space. These are obtained by selecting a centre in feature space based on weighted density, as described above. Two spherical neighbourhoods are then constructed around this centre. Points inside the inner sphere are accepted; points in the spherical shell between the two surfaces are rejected, and the classifier abstains on points outside the outer sphere. The radii of the two surfaces are chosen to minimise the penalty function

$$\tilde{Z} = W_0 + 2\sqrt{W_+ W_-} \tag{10}$$

where notation is as in Section 2.1. This strikes a compromise between classification accuracy and the effective domain of the weak learner [19]. We have found it useful nevertheless to constrain the radius of the outer sphere to be larger than a multiple of the radius of the inner sphere, in order to avoid an excessive localisation of the weak learners.

Using specialist weak learners forces all decisions to be local, which might be advantageous in the case, for instance, of non-convex clusters. From the point of view of the boosting algorithm, these weak learners are treated as specified in Section 2.1

4 Experimental evaluation

The experimental evaluation was based on four well-known real data-sets: the Iris dataset, the Wisconsin breast cancer (referred to as WBC), the Ionosphere data set – all from the USC Machine Learning Repository[4] – and the Biomed dataset[5]. Datasets differ in terms of number of patterns and dimension – details

[4] Available at `http://archive.ics.uci.edu/ml/datasets.html`
[5] Available at `http://lib.stat.cmu.edu/datasets/`

Table 4. Results for Cloosting on different publicly available datasets

Dataset	With Abstention		Without Abstention		K-Means
	RAND-INIT	KM-INIT	RAND-INIT	KM-INIT	
Iris	88.7%	**94.7%**	90.7%	90.7%	89.0%
Biomed	88.6%	88.7%	88.1%	**89.2%**	88.7%
WBC	96.8%	**97.2%**	96.6%	96.6%	96.1%
Ionosphere	**72.9%**	62.1%	69.8%	64.4%	65.2%

may be found in Table 3. All datasets have been standardised before applying the clustering methodology.

We compare two versions of the proposed approach, the first with weak learners that abstain, the second with the original version of the weak learners. Tests with two kinds of initialisation have been carried out: in the former, the initial labels have been chosen using the k-means algorithm (KM-INIT), while in the latter they have been randomly selected (RAND-INIT). In order to be robust against initialisation-driven fluctuations (also the k-means algorithm starts from a random assignment), all experiments have been repeated 5 times and the best result has been taken.

A preliminary evaluation of the impact of the choice of the parameters has shown that the number of Adaboost iterations T, the regularization parameter C of Adaboost-REG and the constants γ and δ for calculating the data density in subtractive weak learners are fairly insensitive; in all experiments they have been fixed to $T = 15$, $C = 20$ and $\gamma = 1.5$ respectively, with $\delta = 3\gamma/4$. On the contrary, the constant β of the smoothing kernel in Equation 2 of Table 2 and (limited to weak learners that abstain) the minimum ratio between the outer sphere and the inner sphere of the learner have proved to be dataset dependent, so that a different value has been used in each case.

Since true labels are known, clustering accuracies can be quantitatively assessed. In particular, given a specific group, an error is considered when a pattern does not belong to the most frequent class inside the group (following the protocol of [3]). The results obtained for each dataset with the different versions of the algorithm are displayed in Table 4. As a reference, in the last column, we report the results obtained with the standard k-means (for the sake of fairness, also in this case we performed 5 random initialisations, picking the best result).

As can be inferred from the table, Cloosting appears to work rather well on these datasets, that are heterogeneous in terms of the number of patterns and of dimensionality of the feature space. Concerning the different versions of the method, the use of weak learners that abstain seems to increase performance. Moreover, it may be noted that k-means initialisation seems to permit a more accurate clustering than random initialisation. Interestingly this is not true for the Ionosphere database, on which k-means performs rather poorly. In this case our method works better if started from random assignments. Finally it may be noted that Cloosting compares favourably with the k-means algorithm (for two

Table 5. Comparative results for three datasets. We refer the reader to the cited papers for details of the experimental protocols.

Iris	
Self organising maps (SOM) [12]	81.0
Neural gas [13]	91.7
Spectral clustering [15]	84.3
Gaussian Mixture Models [14]	89.3
Kernel clustering [3]	**94.7**
Soft kernel clustering [2]	93.3
Cloosting	**94.7**

WBC	
Self organising maps (SOM) [12]	96.7
Neural gas [13]	96.1
Spectral clustering [15]	95.5
Gaussian Mixture Models [14]	94.6
Kernel clustering [3]	97.0
Soft kernel clustering [2]	97.1
Cloosting	**97.2**

Biomed	
Kernel clustering [3]	83.0
Soft kernel clustering [2]	88.2
Cloosting	**89.2**

datasets the improvement is significant), especially when the dimension of the space is rather large (as in the Ionosphere case, dimensionality 34).

As a further comparison with the literature, in Table 5 we report state of the art results[6] for some of the datasets – we also include our best result for comparison. It may be noticed that the performances of Cloosting are in line with those of other state of the art methods.

A final consideration concerns the convergence of Cloosting. Although we do not yet have a theoretical justification for the convergence of the iterative clustering algorithm, in all our experiments the method always converged; the number of iterations required was, on average, around ten to fifteen, depending of course on the initialisation (for k-means initialisation convergence was clearly faster).

5 Conclusions

In this work we take a novel look at clustering from a boosting perspective. Each cluster is modelled by a regularised boosted classifier composed of weak

[6] Some of the results have been computed by the authors, some others have been taken from [3] and [2].

learners inspired by Subtractive Clustering, that can be localised to different areas of feature space. Learning follows a cycle of iterations similar to k-means; a smoothing kernel is used to stabilise the assignment of points to the various clusters.

The convergence properties of the algorithm are at this early stage established empirically, and a smart parameter setting policy still needs to be designed. However, the experiments we report show that Cloosting achieves optimal results on several standard datasets, either matching or improving the performance of state of the art algorithms. This proves, in our view, the potential of the technique and justifies further inquiry into the use of boosting for modelling clusters.

Acknowledgements

Most of the present work was done while F. Smeraldi was visiting professor at PLUS Lab, Italian Institute of Technology, Genoa; F.S. gratefully acknowledges the financial support of IIT.

References

1. Avidan, S.: Spatialboost: adding spatial reasoning to Adaboost. In: Proceedings of ECCV. vol. IV, pp. 386–396 (2006)
2. Bicego, M., Figueiredo, M.: Soft clustering using weighted one-class support vector machines. Pattern Recognition 42(1), 27–32 (2009)
3. Camastra, F., Verri, A.: A novel kernel method for clustering. IEEE Trans. on Pattern Analysis and Machine Intelligence 27, 801–805 (2005)
4. Chiu, S.: Fuzzy model identification based on cluster estimation. Journal of intelligent and fuzzy systems 2, 267–278 (1994)
5. Chiu, S.L.: In: D. Dubois, H. Prade and R. Yager Eds, Fuzzy Information Engineering: A guided tour of applications, chap. 9. John Wiley & Sons (1997)
6. Demiriz, A., Bennet, K.P., Shawe-Taylor, J.: Linear programming boosting via column generation. Machine Learning 46, 225–254 (2002)
7. Freund, Y.: An adaptive version of the boost by majority algorithm. In: Proceedings of the Twelfth Annual Conference on Computational Learning Theory. pp. 102–113 (2000)
8. Freund, Y., Schapire, R.: A decision-theoretic generalization of on-line learning and an application to boosting. Journal of Computer and System Science 55(1) (1997)
9. Freund, Y., Schapire, R.: A short introduction to boosting. Journal of Japanese society for Artificial Intelligence Science 14(5) (1999)
10. Frossyniotis, D., Likas, A., Stafylopatis, A.: A clustering method based on boosting. Pattern Recognition Letters 25, 641–654 (2004)
11. Kim, T.K., Cipolla, R.: MCBoost: multiple classifer boosting for perceptual co-clustering of images and visual features. In: Advances in Neural Information Processing Systems, Vancouver (Canada). pp. 841–856 (2008)
12. Kohonen, T.: Self-Organizing Maps. Springer-Verlag (1997)
13. Martinetz, T., Schulten, K.: Neural-gas network for vector quantization and its application to time-series prediction. IEEE Trans. on Neural Networks 4(4), 558–569 (1993)

14. McLachlan, G., Peel, D.: Finite mixture models. John Wiley and Sons (2000)
15. Ng, A., Jordan, M., Weiss, Y.: On spectral clustering: Analysis and an algorithm. In: Advances in Neural Information Processing Systems. pp. 849–856 (2001)
16. Nock, R., Nielsen, F.: On weighting clustering. IEEE Transactions on PAMI 28(8), 1223–1235 (2006)
17. Press, W.H., Teukolsky, S.A., Vetterling, W.T., Flannery, B.P.: Numerical recipes in C++, chap. 10. Cambridge University Press (2002)
18. Rätsch, G., Onoda, T., Muller, K.R.: Soft margins for adaboost. Machine Learning 42(3), 287–320 (2001), citeseer.ist.psu.edu/657521.html
19. Schapire, R., Singer, Y.: Improved boosting algorithms using confidence-rated predictions. Machine Learning 37(3) (1999)
20. Topchy, A., Minaei-Bidgoli, B., Jain, A.K., Punch, W.F.: Adaptive clustering ensembles. In: Proc. 17th Intl Conf. Pattern Recognition. pp. 272–275 (2004)
21. Warmuth, M., Glocer, K.A., Rätsch, G.: Boosting algorithms for maximizing the soft margin. In: Advances in Neural Information Processing Systems 20, pp. 1585–1592. MIT Press (2008)
22. Warmuth, M.K., Glocer, K.A., Vishwanathan, S.: Entropy regularized lpboost. In: Proceedings of the 19th international conference on Algorithmic Learning Theory (ALT'08). pp. 256–271 (2008)