A robust semi-supervised EM-based clustering algorithm with a reject option

Christophe Saint-Jean, Carl Frélicot

A robust semi-supervised EM-based clustering algorithm with a reject option

Christophe Saint-Jean and Carl Frélicot
L3I - UPRES EA 2118
Avenue Michel Crépeau, 17042 La Rochelle Cedex 1, France
{csaintje,cfrelico}@univ-lr.fr

Abstract

In this paper, we address the problem of semi-supervision in the framework of parametric clustering by using labeled and unlabeled data together. Clustering algorithms can take advantage from few labeled instances in order to tune parameters, improve convergence and overcome local extrema due to bad initialization. We extend a robust parametric clustering algorithm able to manage outlier rejection to the semi-supervision approach. This is achieved by modifying the Expectation-Maximization algorithm. The proposed method shows good performance with respect to data structure discovering, even facing to outliers.

1. Introducing partial supervision in clustering algorithms

Clustering is an important task for exploratory data analysis. It aims at searching for structure in data on the basis on some similarity measure. This process can be done through an extremum search for an appropriate objective function, e.g Fuzzy C-Means (FCM) algorithm in [5] or Expectation-Maximization (EM) approach in [7]. However, the complexity of space to be observed generally imposes the use of sub-optimal methods leading to local extrema. Thus, a good choice of parameter values and especially of initial conditions are essential for the method success.

Partial supervision occurs when both unsupervised and supervised examples are available. Unsupervised data are generally easy to obtain whereas supervised one can need a costly expertise. The main idea of partial supervision is to increase the estimation accuracy of classes parameters and avoid local extrema of the objective function. Let us illustrate how semi-supervision can help to overcome these problems on two examples. Left part of Fig. 1 shows the clustering result of two relatively well-separated groups of points and some outlying points using the FCM algorithm under poor initialization. It fails because of an attempt to obtain separated clusters instead of compact ones. Clusters provided by the CEM algorithm [6] on a XOR-type configuration are shown in Fig. 1 (right part). Here again, data structure is not recovered despite 50 random initializations (the best result is shown). Specifying appropriate points as supervised ones allows to correctly cluster these two data sets. Many popular clustering algorithms have been extended to partial supervision including Fuzzy C-Means in [11], [4], Hierarchical Clustering Algorithms in [1], Point-Prototype Clustering in [3], or Support Vector Machines in [2]. In this paper, we focus on the EM approach which is parametric.

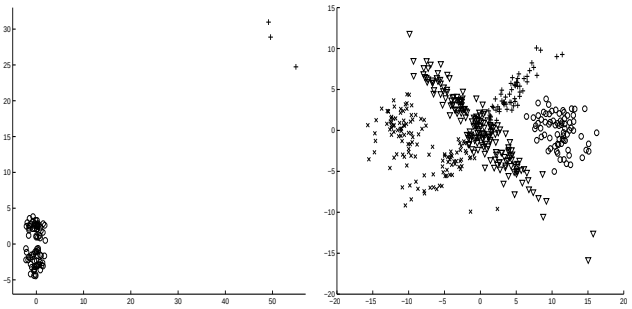

Figure 1. Two examples where a clustering method always fail - FCM (left), CEM (right)

2. A semi-supervision version of parametric clustering with EM

Finite mixtures are commonly used in pattern recognition for unsupervised learning (clustering). From the statistical point of view of clustering, data are assumed to arise from an unknown probability density function f (pdf). Finite mixtures models are a powerful and convenient approach to make an approximate decomposition of f:

$$f(x) = \sum_{k=1}^{c} \pi_k f(x|\theta_k)$$

where π_k and $f(x|\theta_k)$ are respectively the mixing proportions (prior probabilities summing up to one) and the class-conditional densities parameterized by $theta_k$. Note that

we do not address here the problem of the number c of components involved. Let $\mathbf{x} = \{x_1, x_2, \ldots, x_n\}$ be a set of n independent and identically distributed samples from a c-component mixture. The log-likelihood function is

$$log\mathcal{L}(\Theta, \mathbf{x}) = \sum_{i=1}^{n} \log \left(\sum_{k=1}^{c} \pi_k f(x_i|\theta_k) \right) \quad (1)$$

where $\Theta = [\pi_1, \ldots, \pi_{c-1}, \theta_1, \ldots, \theta_c]$ stands for the model parameters to be estimated (notice that π_c is redundant as $\pi_c = 1 - \sum_{k=1}^{c-1} \pi_k$). The estimator $\hat{\Theta}$ that maximizes $\log \mathcal{L}(\Theta, \mathbf{x})$ cannot be found analytically. A possible approach is the EM algorithm which is a general framework to incomplete data problems. From this point of view, an observed data x_i can be regarded as being incomplete where the missing part is the true class labelling. Therefore, one can form complete data $y_i = (x_i, z_i)$ by introducing for all unlabeled examples x_i the realization $z_i = (z_{i1}, \ldots, z_{ic})$ of a c-dimensional random variable Z representing the labels of x_i, i.e. z_{ik} is equal to 1 when x_i arises from the k^{th} component and 0 otherwise. Z is generally supposed to be a multinomial random variable parameterized by $\pi = (\pi_1, \ldots, \pi_c)^T$. Let $\mathbf{z} = \{z_1, x_2, \ldots, z_n\}$ denotes a set of realizations of Z. Under these assumptions, the complete log-likelihood is defined as

$$log\mathcal{L}_c(\Theta, x, z) = \sum_{i=1}^{n} \sum_{k=1}^{c} z_{ik} \log \left(\pi_k f(x_i|\Theta_k) \right) \quad (2)$$

In the context of partial supervision, $log\mathcal{L}_c$ can be split up into unlabeled and labeled part ($\mathbf{x}^u \bigcup \mathbf{x}^d = \mathbf{x}$):

$$\begin{aligned} log\mathcal{L}_c(\Theta, x, z) &= \underbrace{\sum_{x_i \in \mathbf{x}^u} \sum_{k=1}^{c} z_{ik} \log \left(\pi_k f(x_i|\Theta_k) \right)}_{\mathbf{x}^u \text{ unlabeled examples}} \\ &+ \underbrace{\sum_{x_i \in \mathbf{x}^d} \sum_{k=1}^{c} z_{ik} \log \left(\pi_k f(x_i|\Theta_k) \right)}_{\mathbf{x}^d \text{ labeled examples}} \quad (3) \end{aligned}$$

The EM algorithm is a hill-climbing procedure that finds a local maximum of $log\mathcal{L}_c(\Theta, x, z)$. It proceeds in two steps:

- **E-Step** : Computation of the conditional expectation of $log\mathcal{L}_c$, given $\mathbf{x}$ and current parameter estimates $\hat{\Theta}^{(t)}$:

$$Q(\Theta|\hat{\Theta}^{(t)}) \equiv E[log\mathcal{L}_c(\Theta, \mathbf{x}, \mathbf{z})|\mathbf{x}, \hat{\Theta}^{(t)}] \quad (4)$$

  As $Q(\Theta|\Theta^{(t)})$ is a linear combination of $\hat{z}_{ik}$, we just have to compute

$$\hat{z}_{ik} = E[z_{ik}|x_i, \hat{\Theta}^{(t)}] = \frac{\hat{\pi}_k f(x_i|\hat{\Theta}_k)}{\sum_{l=1}^{c} \hat{\pi}_l f(x_i|\hat{\Theta}_l)} \quad (5)$$

  for all unlabeled examples. For supervised ones, z_i are not estimated but fixed by the expert (possibly not binary).

- **M-Step** : Update parameters estimates to maximize $Q(\Theta|\Theta^{(t)})$

$$\hat{\Theta}^{(t+1)} = arg \max_{\Theta}(Q(\Theta|\hat{\Theta}^{(t)})) \quad (6)$$

  This computation depends on the theoretical model for classes to be tracked. We propose to use a model that is robust enough to cope with outliers.

3. A robust approach with a reject option

3.1. Theoretical model for classes

When a normal mixture model is assumed, the parameters Θ_k of each component are the mean μ_k and the covariance matrix Σ_k of cluster C_k. In [12], we have proposed to model each cluster C_k as a mixture of two normal subcomponents:

$$f(x|C_k) = \underbrace{(1 - \gamma_k)\mathcal{N}(x; \mu_k, \Sigma_k)}_{(A)} + \underbrace{\gamma_k \mathcal{N}(x; \mu_k, \alpha_k \Sigma_k)}_{(B)} \quad (7)$$

First term (A) intends to track cluster kernel points while second term (B) allows to take into account surrounding outliers via a multiplicative coefficient. Parameters γ_k and α_k control respectively the combination of the two subcomponents and the spread of the second one by modifying its covariance. Parameters γ_k has to be taken in $[0, 0.5]$ to penalize (B) and α_k must be greater than 1. This model corresponds to an ϵ-contamination assumption which is the basis of the Student t-distribution construction (see [10],p. 223).

3.2. Estimation of classes parameters

The key point of the approach we propose is the use of different estimates for the same random variables μ_k, Σ_k for (A) and for (B). In the following, $\tilde{\mu}_k$, $\tilde{\Sigma}_k$ stand for robust estimates (A) whereas $\hat{\mu}_k$, $\hat{\Sigma}_k$ denotes standard estimates (B). The principle of this approach is to allow standard estimates to be disturbed by outliers while robust estimates concentrate on class-kernel parameters.
Robust estimates are provided through an iterative procedure, quite similar to the reweighted least-squares, that uses robust M-estimates [8]. Such an estimator is an influence function where the weight w_i associated to the i^{th} sample x_i decreases as its distance to the cluster prototype increases (e.g Mahalanobis distance $d^2_{\mathcal{M}} = (x - \mu^k)^T \Sigma_k^{-1} (x - \mu_k)$). The area of influence is controlled by a threshold, say h: the

larger h, the more samples are involved in the estimation process. Fig. 2 shows how weight w_i is related to distance $d_{\mathcal{M}}$ and threshold h of the so-called Huber M-estimator. Algorithm 1 describes the whole estimation process replacing parameters updating equations involved during the M-Step of classical EM.

H 1: Parameters update (M-Step)

Input: $\mathbf{x} = \{x_1, \ldots, x_n\}, \hat{z}_{ik}$ current estimates of $P(C_k|x_i)$, *h* the M-estimator threshold having ψ as influence function

$\tau \leftarrow 0$;

$\hat{\pi}_k^{(t+1)} = \frac{\sum_{x_i \in \mathbf{x}^u} \hat{z}_{ik}^{(t)} + \sum_{x_i \in \mathbf{x}^d} \hat{z}_{ik}}{card(\mathbf{x}^u) + card(\mathbf{x}^d)}$

$\tilde{\mu}_{k,0} = \hat{\mu}_k^{(t+1)} = \frac{\sum_{x_i \in \mathbf{x}^u} \hat{z}_{ik}^{(t)} x_i + \sum_{x_i \in \mathbf{x}^d} z_{ik} x_i}{\sum_{x_i \in \mathbf{x}^u} \hat{z}_{ik}^{(t)} + \sum_{x_i \in \mathbf{x}^d} z_{ik}}$;

$\tilde{\Sigma}_{k,0} = \hat{\Sigma}_k^{(t+1)} = \frac{\sum_{x_i \in \mathbf{x}^u} \hat{z}_{ik}^{(t)} \tilde{S}_{ik,0} + \sum_{x_i \in \mathbf{x}^d} z_{ik} \tilde{S}_{ik,0}}{\sum_{x_i \in \chi^u} \hat{z}_{ik}^{(t)} + \sum_{x_i \in \mathbf{x}^d} z_{ik}}$;

with $\tilde{S}_{ik,0} = (x_i - \tilde{\mu}_{k,0})(x_i - \tilde{\mu}_{k,0})^T$;

repeat

  $\tau \leftarrow \tau + 1$;

  for $x_i \in \mathbf{x}^u \bigcup \mathbf{x}^d$ **do**

    $e_{i,\tau} = d_{\mathcal{M}}(x_i; \tilde{\mu}_{k,\tau-1}, \tilde{\Sigma}_{k,\tau-1})$;

    $w_{i,\tau} = \frac{\psi(e_{i,\tau}, h)}{e_{i,\tau}}$;

  $\tilde{\mu}_{k,\tau} = \frac{\sum_{x_i \in \mathbf{x}^u} w_{i,\tau} \hat{z}_{ik}^{(t)} x_i + \sum_{x_i \in \mathbf{x}^d} w_{i,\tau} z_{ik} x_i}{\sum_{x_i \in \mathbf{x}^u} w_{i,\tau} \hat{z}_{ik}^{(t)} + \sum_{x_i \in \mathbf{x}^d} w_{i,\tau} z_{ik}}$;

  $\tilde{\Sigma}_{k,\tau} = \frac{\sum_{x_i \in \mathbf{x}^u} w_{i,\tau} \hat{z}_{ik}^{(t)} \tilde{S}_{ik,\tau} + \sum_{x_i \in \mathbf{x}^d} w_{i,\tau} z_{ik} \tilde{S}_{ik,\tau}}{\sum_{x_i \in \mathbf{x}^u} w_{i,\tau} \hat{z}_{ik}^{(t)} + \sum_{x_i \in \mathbf{x}^d} w_{i,\tau} z_{ik}}$;

  with $\tilde{S}_{ik,\tau} = (x_i - \tilde{\mu}_{k,\tau})(x_i - \tilde{\mu}_{k,\tau})^T$;

until *Stop Criterion*;

$\tilde{\mu}_k^{(t+1)} = \tilde{\mu}_{k,\tau}$;

$\tilde{\Sigma}_k^{(t+1)} = \tilde{\Sigma}_{k,\tau}$;

Index t refers to the current M-Step while τ refers to the robust one. The more τ, the less points are used in the robust estimation process. One can need to stop this loop by using a criterion involving an upper bound on either τ or κ the rate of samples having a quite zero weight.

It can be shown that the property of monotonous increase of log-likelihood (see Eq. 1) no more holds because of the iterative estimation process which results in approximating instead of strictly maximizing it.

We propose to update threshold h of the Huber M-estimator during the iterative estimation with the help of labeled samples, e.g. by taking the maximum distance between supervised points and the mean estimates of their true class :

$$h_k = \max_{x_i \in \mathbf{x}^d} (d_{\mathcal{M}}(x_i, C_k), h_{min}) \tag{8}$$

where h_{min} denotes a minimum value insuring sufficient statistics.

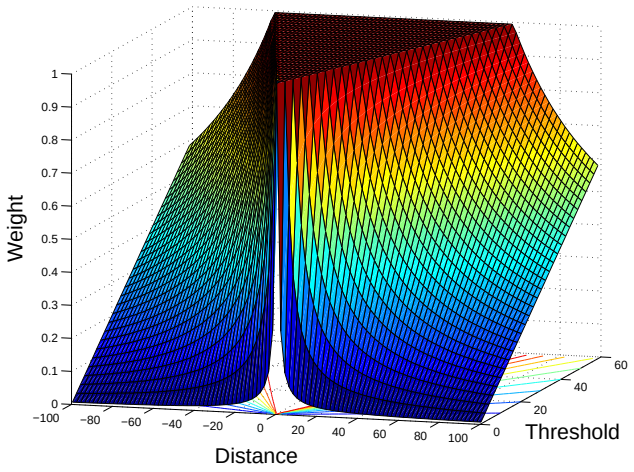

Figure 2. Huber M-estimator: weight as a function of distance and threshold h

3.3. Decision rule and outliers rejection

Outlier rejection is performed during the affectation step by the way of an additional $(c{+}1)^{th}$ cluster, called the reject cluster, which is created by summing all (B)-terms. Then, the mixture decomposition is as follows:

$$\begin{aligned} \hat{f}(x) &= \underbrace{\sum_{k=1}^{c} \hat{\pi}_k \left[(1-\gamma_k)\, \mathcal{N}(x; \tilde{\mu}_k, \tilde{\Sigma}_k) \right]}_{c \text{ clusters}} \\ &+ \underbrace{\sum_{k=1}^{c} \hat{\pi}_k \left[\gamma_k\, \mathcal{N}(x; \hat{\mu}_k, \alpha_k \hat{\Sigma}_k) \right]}_{\text{rejection class}} \end{aligned} \tag{9}$$

Affectation is carried out according the Maximum A Posteriori criterion (MAP) among the $(c{+}1)$ clusters:

$$\hat{P}(C_k|x) = \frac{\hat{\pi}_k (1-\gamma_k)\, \mathcal{N}(x; \tilde{\mu}_k, \tilde{\Sigma}_k)}{\hat{f}(x)} \tag{10}$$

$$\hat{P}(Reject|x) = \frac{\sum_{k=1}^{c} \hat{\pi}_k \gamma_k\, \mathcal{N}(x; \hat{\mu}_k, \alpha_k \hat{\Sigma}_k)}{\hat{f}(x)} \tag{11}$$

4. Experiments

The first result we present deals with the two artificial data sets shown in Fig. 1 which structure is not correctly detected when totally non robust or non supervised algorithms are used. Fig. 3 shows the clusters obtained with the method we propose. Parameters γ_k and α_k are randomly chosen in the intervals mentioned in the previous section. For the left case 4.82% of points are supervised (•-marked) while

3.1% for the right one. In both cases, the proposed method provide clusters that look good and correctly reject outliers (□-marked). We have experimented the proposed method

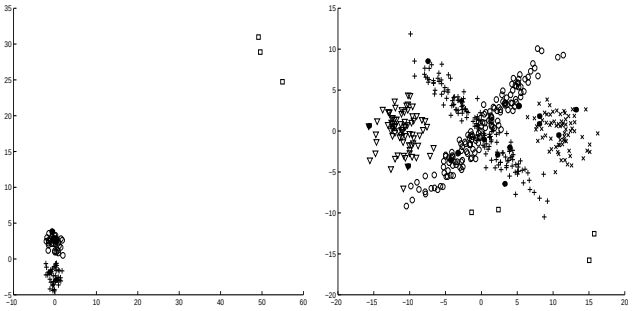

Figure 3. Two examples where the proposed method success

on well-known real data sets from the UCI repository[1] with various random settings of parameters γ_k in $[0.05, 0.5]$, α_k in $[1, 20]$, up to 50 different initializations and supervision rate s varying from 0% (no supervision) to 100% (total supervision).

Some particular results are reported in Table 1. In order to compare with other published results, e.g. by [9] (partial supervision without rejection), we make a resubstitution according to the MAP criterion. As expected, a low s is generally sufficient to significantly reduce error rates P_e and to significantly increase success rates P_c without making the reject rates P_r too high.

More generally, the obtained results are comparable with those of supervised methods[2]. From Table 1, it is worthy of note that P_c can be used as a validation index when setting $s = 100\%$, e.g. *Wine* data are known to be easily separable while *Pima Indian Diabetes* data are not. For the *Iris* data, it is fruitless to try to improve P_c over a particular low value of s. In [9], an error rate of 3.33% has been obtained for a supervision rate of 10% and taking original classes proportions into account while selecting supervised points. Without this assumption, we divide this error by more than two (1.33%) for about twice less supervision (5.33%).

5. Conclusion

In this article, the advantage of the semi-supervision in context of clustering multivariate data is recalled. We propose a new algorithm based on EM which differs from the classical approach by introducing of a reject option that makes it possible to deal with outliers, as well as the use of a robust M-estimator. We focussed here on the Huber M-estimator, but others can be used. Experiments results obtained on various artificial and real datasets show that the method achieve performance equivalent or better than other semi-supervised clustering approaches.

[1] http://www.ics.uci.edu/mlearn/MLRepository.html

[2] http://www.phys.uni.torun.pl/kmk/projects/datasets.html

Dataset	s	P_c	P_e	P_r
Pima Indians Diabetes N=768,C=2,P=8	0.%	66.93%	29.04%	4.04%
	6.77%	70.18%	29.82%	0.%
	100.%	76.56%	23.44%	0.%
Iris N=150,C=3,P=4	0.%	96.67%	2.66%	0.67%
	5.33%	98.66%	1.33%	0.%
	100.%	98.66%	1.33%	0.%
Wine N=178,C=3,P=13	0.%	89.32%	5.62%	5.06%
	1.78%	96.07%	1.12%	2.81%
	100.%	100.%	0.%	0.%

Table 1. Real datasets - Selected results

References

[1] A. Amar, N. Labzour, and A. Bensaid. Semi-supervised hierarchical clustering algorithms. In *Sixth Scandinavian Conference on Artificial Intelligence*, pages 232–239, Helsinki, Finland, August 18-20 1997.

[2] K. Bennett and A. Demiriz. Semi-supervised support vector machines. *Advances in Neural Information Processing Systems*, 11:368–374, 1999.

[3] A. Bensaid and J. Bezdek. Semi-supervised point-prototype clustering. *Int'l Journal of Pattern Recognition and Artificial Intelligence*, 12(5), 1998.

[4] A. Bensaid, L. Hall, J. Bezdek, and L. Clarke. Partially supervised clustering for image segmentation. *Pattern Recognition*, 29:859–871, 1996.

[5] J. Bezdek. *Pattern Recognition with Fuzzy Objective Function Algorithms*. Plenum Press, New York, (second edition) edition, 1987.

[6] G. Celeux and G. Govaert. A classification EM algorithm for clustering and two stochastic versions. *Computational Statistics and Data Analysis*, 14:315–332, 1992.

[7] A. P. Dempster, N. M. Laird, and D. B. Rubin. Maximum likelihood from incomplete data via the EM algorithm (with discussion). *Journal of the Royal Statistical Society series B*, 39:1–38, 1977.

[8] P. J. Huber. *Robust Statistics*. John Wiley, New York, 1981.

[9] N. Labzour, A. Bensaid, and J. Bezdek. Improved semi-supervised point-prototype clustering algorithms. *IEEE Int'l Conf. on Fuzzy Systems , IEEE World Congress on Computational Intelligence*, pages 1383–1387, May 1998.

[10] G. McLachlan and D. Peel. *Finite Mixture Models*. Eds Wiley, 2000. ISBN 0-471-00626-2.

[11] W. Pedrycz and J. Waletzky. Fuzzy clustering with partial supervision. *IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics*, 27(5):787–795, 1997.

[12] C. Saint-Jean, C. Frélicot, and B. Vachon. *Clustering with EM: complex models vs. robust estimation*, pages 872–881. In proceedings of SPR 2000: F. J. Ferri, J. M. Inesta, A. Amin, and P. Pudil (Eds.). Lectures Notes in Computer Science 1876, Springer-Verlag, 2000.