CARP: Software for Fishing Out Good Clustering Algorithms

Volodymyr Melnykov VOLODYMYR.MELNYKOV@NDSU.EDU
Department of Statistics
North Dakota State University
Fargo, ND 58102, USA

Ranjan Maitra MAITRA@IASTATE.EDU
Department of Statistics and Statistical Laboratory
Iowa State University
Ames, IA 50011, USA

Editor: Cheng Soon Ong

Abstract

This paper presents the CLUSTERING ALGORITHMS' REFEREE PACKAGE or CARP, an open source GNU GPL-licensed C package for evaluating clustering algorithms. Calibrating performance of such algorithms is important and CARP addresses this need by generating datasets of different clustering complexity and by assessing the performance of the concerned algorithm in terms of its ability to classify each dataset relative to the true grouping. This paper briefly describes the software and its capabilities.

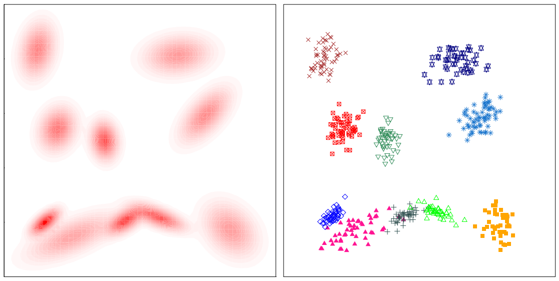
Keywords: CARP, MIXSIM, clustering algorithm, Gaussian mixture, overlap

1. Introduction

There are many clustering algorithms available, but no uniformly clear winner. Thus, calibrating the performance of each algorithm in different situations is important. CARP provides software to evaluate performance of any user-provided clustering technique under simulated datasets of specified clustering complexity. At its heart is software that implements Maitra and Melnykov's (2010) algorithm to simulate datasets from Gaussian mixtures of different clustering difficulty. CARP provides an integrated software tool which generates datasets using the above, uses the clustering algorithm being evaluated on each dataset and compares the derived grouping relative to the true via indices such as the Adjusted Rand ($\mathcal{R}$) index (Hubert and Arabie, 1985). This paper discusses usage, applicability and flexibility of CARP.

2. CARP: An Integrated Tool for Evaluating Clustering Algorithms

There is some software available for simulating datasets for evaluating performance of clustering algorithms, for example, Qiu and Joe's (2006) open-source R package CLUSTERGENERATION (formerly, GENCLUS) and Steinley and Henson's (2005) publicly unavailable OCLUS code for use only in the proprietary MATLAB package. Beyond the algorithmic and other limitations (Maitra and Melnykov, 2010) underlying both these packages, none of them provide an integrated tool to evaluate clustering algorithms. CARP addresses this shortcoming by seamlessly integrating three stages. The first stage generates datasets given user-specified measures for desired clustering complexity

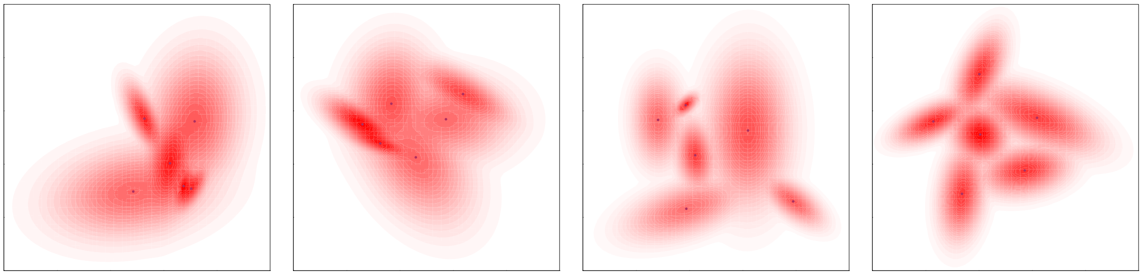


(a) (b)

Figure 1: (a) Sample 10-component bivariate mixture distribution. (b) Corresponding sample dataset generated at the first stage of CARP.

following the definitions of Maitra and Melnykov (2010). The next stage clusters each dataset using the user-provided algorithm in executable form. The final phase evaluates performance—in terms of the default $\mathcal{R}$ or some other index supplied by the user in executable form—of this clustering algorithm by comparing its derived partitioning with the true grouping. The package is specifically designed to be flexible enough to allow the user to provide clustering algorithms and evaluation measures in his/her preferred programming language. We now detail the three stages of CARP.

Stage I: Simulating Datasets of Given Clustering Complexity: The first stage of CARP implements Maitra and Melnykov's (2010) algorithms to generate datasets from Gaussian mixtures of different numbers of components, dimensions and dispersions characterized through summaries of the pairwise overlap that serves as a surrogate measure for clustering complexity. For any two components (say i, j) of a Gaussian mixture density $g(x) = \sum_{k=1}^{K} \pi_k \phi(x; \mu_k, \Sigma_k)$, the overlap is defined as $\omega_{ij} = \omega_{i|j} + \omega_{j|i}$, where $\omega_{j|i}$ is the probability that an observation from the ith component is misclassified to be from the jth one, with $\omega_{i|j}$ defined similarly. Analytical expressions for $\omega_{i|j}$ and $\omega_{j|i}$ being impractical, we use Theorem 2.1 in Maitra and Melnykov (2010) to calculate them numerically and efficiently. For a K-component mixture, there are $\binom{K}{2}$ ω_{ij}s, so clustering complexity is characterized using the average ($\bar{\omega}$) or maximum ($\check{\omega}$) pairwise overlap measures. The software implements Algorithm 2.2.1 of Maitra and Melnykov (2010) to simulate Gaussian mixtures corresponding to a given $\bar{\omega}$ or $\check{\omega}$, as in the sample ten-component bivariate mixture distribution satisfying $\check{\omega} = 0.1$ of Figure 1a. For a more comprehensive summary of clustering complexity, CARP uses their Algorithm 2.2.2 to simulate mixtures according to the pair $(\bar{\omega}, \check{\omega})$, as illustrated in Figure 2. Note that Figures 2a-b have the same $\bar{\omega}$ but in the first case, this value is driven by the overlap between a few pairs of components. In the second case, many more components contribute. Similarly



(a) $\bar{\omega} = 0.05$, $\check{\omega} = 0.198$ (b) $\bar{\omega} = 0.05$, $\check{\omega} = 0.135$ (c) $\bar{\omega} = 0.001$, $\check{\omega} = 0.0036$ (d) $\bar{\omega} = 0.001$, $\check{\omega} = 0.0027$

Figure 2: Sample 6-component bivariate mixture distributions for different $(\bar{\omega}, \check{\omega})$s.

p	1	2	3	5	10	20	50	100
$K = 10$	12.051	8.993	5.360	0.635	0.147	0.206	0.774	4.202
$K = 100$	1236.965	1003.951	574.208	67.567	13.668	19.978	71.023	373.375

Table 1: Median time (secs) to simulate 25 K-component p-dimensional Gaussian mixtures.

for Figures 2c-d. CARP can simulate mixture models of many components (K) and dimensions (p): Table 1 summarizes the median time over 25 samples to obtain realizations of Gaussian mixtures at each setting on a desktop workstation with dual quad-core Intel® Xeon® X5482 @ 3.20 GHz processors running the Fedora 11 Linux distribution with the `2.6.30.9-102.fc11.x86_64` kernel and the `gcc` 4.4.1 suite of compilers. Note that an increase in K affects running time more than p.

The first phase of CARP thus involves simulating mixture distributions corresponding to user-specified K, p and clustering complexity in the form of $\breve{\omega}$ and/or $\bar{\omega}$. Given the desired sample size (n) of the dataset, this stage then obtains n simulated observations from each realized mixture model, as in the 500-observations sample displayed in Figure 1b. Beyond Gaussian mixtures, datasets from more general-shaped clusters with desired approximate clustering complexity are possible to generate using the inverse multivariate Box-Cox transform on the simulated Gaussian mixtures. This stage is also designed to allow for generating datasets with noisy variables and/or containing scatter/outliers. CARP can be used standalone, without calls to the next two stages, to generate only these distributions and datasets if so desired—the R package MIXSIM provides an additional interface to this stage of CARP.

Stage II: Clustering Simulated Datasets: The second phase of CARP uses the clustering algorithm(s) being evaluated to partition the datasets simulated in the first stage. Code for these algorithm(s) is submitted by the user in executable form, with no requirement on this code having to be in a specific programming language. This stage is designed to interface easily with other clustering tools, as illustrated in the manual.

Stage III: Evaluating the Performance of Clustering Algorithms: The third stage compares the partitionings provided by each clustering algorithm under investigation to the true. The default evaluation metric is $\mathcal{R}$, but other measures, including those with user-supplied code in executable form, can be used: see the manual for examples.

In summary, CARP provides distributions of the desired performance measure for the clustering method(s) being evaluated for given n, K, p and the desired $\breve{\omega}$, $\bar{\omega}$ or $(\bar{\omega}, \breve{\omega})$.

2.1 Demonstrating the Utility of CARP

For illustration, we consider a simple example where the goal is to evaluate performance of different algorithms in partitioning datasets with $(n, K, p) = (100, 7, 5)$ and varying clustering complexity characterized solely in terms of $\breve{\omega}$. (The exact CARP commands used to conduct this study are in Section 5 of the CARP software manual.) Table 2 summarizes performance of each algorithm on 25 datasets, each with different $\breve{\omega}$ and obtained using Stage I of CARP. Clustering algorithms are used here via external calls to the corresponding software. Evaluations are in terms of the default $\mathcal{R}$ provided with CARP. Clearly, performance degrades across the board with higher $\breve{\omega}$. Hierarchical clustering with Ward's and single linkages are, respectively, the best and worst performers in many cases, with k-means being second-best. Although a small-scale experiment, it highlights CARP's utility in summarizing and evaluating performance of different clustering algorithms.

	$\breve{\omega}$	0.500	0.400	0.300	0.200	0.150	0.100	0.050	0.010	0.005	0.001
W	$\mathcal{R}_{0.5}$	0.082	0.199	0.385	0.582	0.715	0.793	0.931	0.980	1.000	1.000
	$I_{\mathcal{R}}$	0.061	0.146	0.197	0.300	0.210	0.161	0.105	0.044	0.024	0.000
C	$\mathcal{R}_{0.5}$	0.063	0.164	0.327	0.477	0.596	0.714	0.787	0.956	0.991	1.000
	$I_{\mathcal{R}}$	0.052	0.125	0.216	0.257	0.231	0.216	0.182	0.145	0.052	0.005
A	$\mathcal{R}_{0.5}$	0.015	0.098	0.227	0.433	0.564	0.689	0.812	0.958	0.990	1.000
	$I_{\mathcal{R}}$	0.022	0.176	0.242	0.205	0.261	0.179	0.133	0.141	0.111	0.000
S	$\mathcal{R}_{0.5}$	0.002	0.003	0.005	0.017	0.058	0.234	0.490	0.820	0.833	0.897
	$I_{\mathcal{R}}$	0.007	0.007	0.010	0.202	0.321	0.442	0.449	0.116	0.148	0.170
K	$\mathcal{R}_{0.5}$	0.086	0.230	0.388	0.621	0.671	0.821	0.912	0.978	0.992	1.000
	$I_{\mathcal{R}}$	0.076	0.149	0.212	0.201	0.212	0.122	0.094	0.029	0.051	0.000
P	$\mathcal{R}_{0.5}$	0.083	0.209	0.341	0.560	0.677	0.811	0.901	0.977	0.982	1.000
	$I_{\mathcal{R}}$	0.059	0.147	0.237	0.208	0.217	0.192	0.102	0.043	0.027	0.000

Table 2: Performance of hierarchical algorithms with Ward's (W), complete (C), average (A), single (S) linkages, *k*-means (K), and partitioning around medoids (P) algorithms on clustering 5-dimensional datasets of size 100 with 7 groups. $\mathcal{R}_{0.5}$ and $I_{\mathcal{R}}$ represent the median and inter-quartile range of $\mathcal{R}$ over 100 replications.

2.2 Implementation

CARP is implemented in ANSI/ISO-compliant C and also available at http://www.mloss.org. Complete details on usage, parameters and examples are provided in the package's manual and README file. CARP can be used standalone in order to only simulate Gaussian mixtures and corresponding datasets. This standalone functionality is also provided by the R package MIXSIM which is publicly available from http://www.R-project.org.

3. Conclusions

CARP is a powerful and user-friendly open source software package for evaluating clustering algorithms. It realizes mixture models of pre-specified clustering complexity, and from there, datasets of desired sample sizes that are then partitioned using the clustering algorithms being appraised. Performance is summarized by comparing the obtained groupings with the true. CARP can also be used to evaluate semi-supervised clustering algorithms, or on simulated datasets with noisy variables or containing scatter/outliers. Clusters should each ideally be Gaussian-distributed, though the package also uses transformations such as the multivariate Box-Cox to simulate grouped datasets from more general distributions. CARP can also calculate the pairwise overlap between identified groups in clustered or classified datasets. It is only designed for cases involving continuous variables and not, in general, able to evaluate algorithms that cluster on manifolds and the like.

Acknowledgments

We acknowledge partial support for this project from the National Science Foundation (NSF) CAREER grant #DMS-0437555.

References

L. Hubert and P. Arabie. Comparing partitions. *Journal of Classification*, 2:193–218, 1985.

R. Maitra and V. Melnykov. Simulating data to study performance of finite mixture modeling and clustering algorithms. *Journal of Computational and Graphical Statistics*, 19(2):354–376, 2010. doi: 10.1198/jcgs.2009.08054.

W. Qiu and H. Joe. Generation of random clusters with specified degree of separation. *Journal of Classification*, 23:315–334, 2006.

D. Steinley and R. Henson. OCLUS: An analytic method for generating clusters with known overlap. *Journal of Classification*, 22:221–250, 2005.