BIOMETRICS , 1–9
2008

Supplement to "Clustering in the Presence of Scatter"

Ranjan Maitra and Ivan P. Ramler

1. The k-clips Objective Function and BIC for Estimating K

Let $\boldsymbol{X}_1, \boldsymbol{X}_2, \ldots, \boldsymbol{X}_n$ be p-variate observations from the distribution given by

$$g(\boldsymbol{x}) = \sum_{k=1}^{K} \zeta_k \left[\frac{1}{\sigma} f\left(\frac{\boldsymbol{x} - \boldsymbol{\mu}_k}{\sigma} \right) \right] + \frac{\zeta_{K+1}}{V}$$

where V is the volume of a bounded region $\mathcal{B}$ with uniform density, $\sigma > 0$ is a scale parameter, ζ_k is an identification parameter, $f(\boldsymbol{y}) = \psi(\boldsymbol{y}'\boldsymbol{y})$ and $\psi(\cdot)$ is a real positive-valued function such that $f(\cdot)$ is a p-variate density. In a hard-clustering context, ζ_k is an indicator function corresponding to the class of the observation. Further, let f in the above correspond to a Gaussian density with common dispersion matrix $\boldsymbol{\Sigma}$. Under this setup, the log-likelihood for the parameters, given the data can be written as $\sum_{i=1}^{n} \{\sum_{k=1}^{K} \zeta_{i,k} \log [\phi_p(\boldsymbol{x}_i; \boldsymbol{\mu}_k, \boldsymbol{\Sigma})] - \zeta_{i,K+1} \log(V)\}$, where $\zeta_{i,k}$ is an indicator function corresponding to the class of i^{th} observation and $\phi_p(\cdot)$ the p-variate Gaussian density function. When $\boldsymbol{\Sigma} = \sigma^2 \boldsymbol{I}$, as in the k-means setup, this log-likelihood can be rewritten as $-\frac{p}{2} \log\left(\sigma^2 + 2\pi\right) \sum_{i=1}^{n} \sum_{k=1}^{K} \zeta_{i,k} - \frac{1}{2\sigma^2} \sum_{i=1}^{n} \sum_{k=1}^{K} \zeta_{i,k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 - \log(V) \sum_{i=1}^{n} \zeta_{i,K+1}$. Thus using $\sum_{i=1}^{n} \sum_{k=1}^{K} \zeta_{i,k} = n^*$ and $\sum_{i=1}^{n} \sum_{k=1}^{K} \zeta_{i,k} \|\boldsymbol{x}_i - \boldsymbol{\mu}_k\|^2 = \mathtt{tr}(\boldsymbol{W}^*)$, maximizing this log-likelihood becomes equivalent to minimizing the objective function.

For the more general case of a hard-clustering of Gaussian densities with common $\boldsymbol{\Sigma}$, we get a similar objective function given by $-\frac{n^* p}{2n}(1 + \log 2\pi - \log n^*) - \frac{n^*}{n}\left(\frac{1}{2} \log |\boldsymbol{W}^*|\right) - \left(1 - \frac{n^*}{n}\right) \log \hat{V}$. This follows from noting that the exponent in the likelihood takes the maximum value given by $-\sum_{i=1}^{n} \sum_{k=1}^{K} \zeta_{i,k} (\boldsymbol{x}_i - \boldsymbol{\mu}_k)^T \boldsymbol{\Sigma} (\boldsymbol{x}_i - \boldsymbol{\mu}_k) = -n^* p$.

A BIC approach to estimating the number of clusters under this more general log-likelihood function reduces to minimizing $\log\left(k + \frac{1}{p}\right) + \frac{n^* p}{2n}(1 + \log 2\pi - \log n^*) + \frac{n^*}{n}\left(\frac{1}{2} \log |\boldsymbol{W}^*|\right) + \left(1 - \frac{n^*}{n}\right) \log \hat{V}$. Assuming however, that $\frac{n^*}{n}$ is approximately constant, the second term becomes irrelevant so that we get $\log\left(k + \frac{1}{p}\right) + \frac{n^*}{n}\left(\frac{1}{2} \log |\boldsymbol{W}^*|\right) + \left(1 - \frac{n^*}{n}\right) \log \hat{V}$, which we suggest minimizing in order to find the optimal number of clusters. Our extensive empirical evidence indicates that this choice performs better than when we include the additional term $\frac{n^* p}{2n}(1 + \log 2\pi - \log n^*)$ in the minimization of the objective function. Note that while we could have used the restrictive assumption of homogeneous spherical clusters, we use homogeneous clusters of general dispersion structure, following long-established empirical evidence that Marriott's (1971) criterion works better than $tr(\boldsymbol{W})$ on estimating the number of clusters in datasets with groupings with homogeneous dispersion. This last may be a consequence of the fact that most datasets are probably better grouped using densities with homogeneous general dispersions than homogeneous spherical variance-covariance matrices. Finally, as mentioned in the paper, in the absence of scatter, $n* = n$ and $\boldsymbol{W}^* = \boldsymbol{W}$ so that the algorithm reduces to Marriott's criterion.

2. Additional Experimental Evaluations

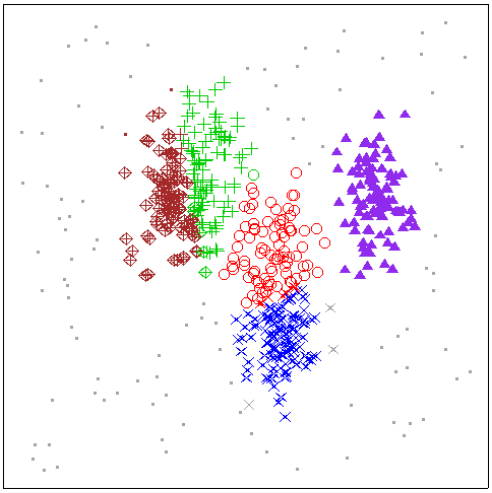
The k-clips algorithm detailed in the paper has three aspects: the algorithm itself, initialization and estimating the number of clusters. Here we evaluate performance of each of these three aspects. The experimental suite is as described in the paper: for $p = 2$, $n = 500$, $s = 100$ and $c = 0.8, 1.2$ or 1.6 and for $p = 5, 10$ and 20 we replicate each experiment 25 times in order to account for simulation variability in our evaluations.

2.1 *Evaluating the* k**-clips** *Algorithm*

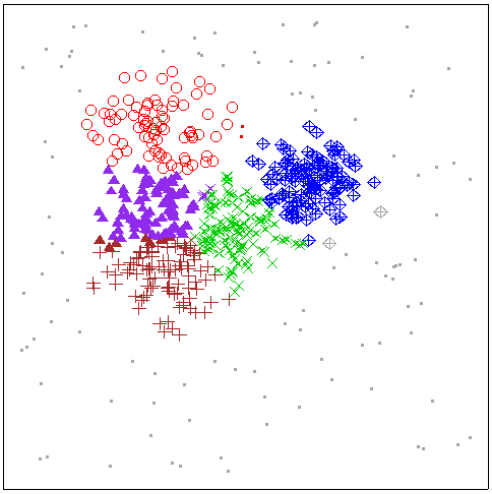
Our first test is on the performance of the algorithm independent of the initialization and given K. We run the algorithm with known K and the true cluster centers as the initializers. For $p = 2$, Figure 1 shows that only a few of the observations along the boundaries of the cores are misclassified. There is a slight improvement as cluster separation increases but in all cases performance is quite good. For the higher dimensional experiments, Table 1 provides summary measures of the adjusted Rand measures over the twenty-five datasets corresponding to different dimensions (p), numbers of clusters (K), amounts of separation (c), and proportion of scatter points (s). We see that the algorithm performs very well, when taking into consideration the degree of scatter, and is very often perfect. Interestingly, the algorithm holds its own for higher dimensions even with high amounts of scatter, but finall has degraded performance for small ($p = 5$) and large (50%) scattered observations. It is encouraging to note that in all cases, the performance of the algorithm improves with increased separation between clusters.

2.2 *Evaluating the Initialization Algorithm*

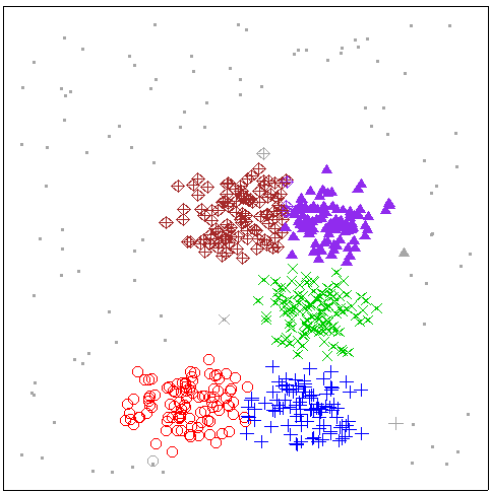
We next tested the initialization scheme for known K. For the bivariate experiments, Figure 2 shows the starting cluster centers in relation to the true cluster centers and true cluster points. In all experiments, the starting centers that are found are located fairly well within actual clusters. The only exception is with $c = 0.8$ where a starting center is located near the edge of a cluster. For the higher dimensional experiments, once the initialization algorithm terminated, the true cluster points among the observations were classified into groups



(a) $c = 0.8$: $\mathcal{R}_a = 0.920$



(b) $c = 1.2$: $\mathcal{R}_a = 0.934$



(c) $c = 1.6$: $\mathcal{R}_a = 0.965$

Figure 1. Results of k-clips independent of initialization algorithm with five clusters given as known for $c = 0.8, 1.2$ and 1.6 for the first, second and third columns respectively. Small filled circles represent identified scatter, true clusters by color and identified clusters by characters.

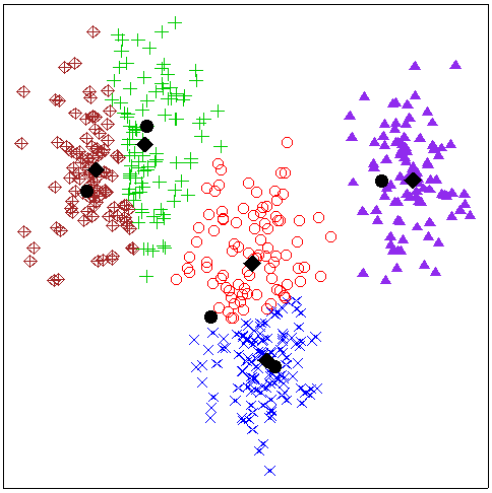
Table 1

Summary of adjusted Rand values of groupings obtained using k-clips with known number of clusters K and true cluster centers as starting centers. The last three columns contain the first quartile ($\mathcal{R}_{\frac{1}{4}}$), median ($\mathcal{R}_{\frac{1}{2}}$) and third quartile ($\mathcal{R}_{\frac{3}{4}}$) of the adjusted Rand values of the twenty-five runs respectively.

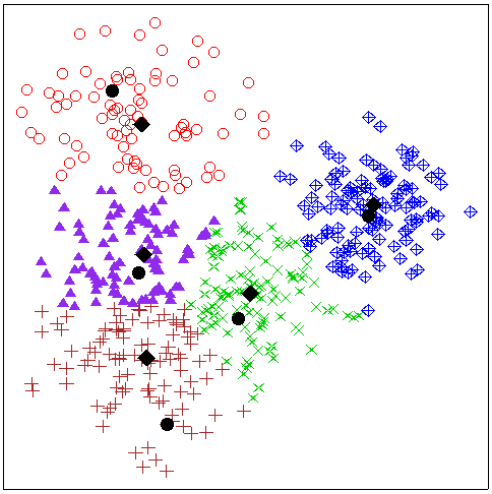
	s	c	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$
$p = 5, K = 5, n = 500$	.15	0.8	0.890	0.921	0.947
		1.2	0.966	0.982	0.989
		1.6	0.994	0.995	0.998
		2.0	0.995	0.997	1.000
	.25	0.8	0.881	0.915	0.944
		1.2	0.933	0.955	0.967
		1.6	0.963	0.978	0.987
		2.0	0.978	0.985	0.990
	.50	2.0	0.902	0.916	0.941
$p = 10, K = 7, n = 2000$	.15	0.8	0.958	0.968	0.976
		1.2	0.995	0.997	0.999
		1.6	0.999	1.000	1.000
		2.0	1.000	1.000	1.000
	.25	0.8	0.959	0.968	0.977
		1.2	0.994	0.996	0.997
		1.6	1.000	1.000	1.000
		2.0	1.000	1.000	1.000
	.50	2.0	0.999	1.000	1.000
$p = 20, K = 15, n = 5000$	.15	0.8	0.996	0.997	0.997
		1.2	1.000	1.000	1.000
		1.6	1.000	1.000	1.000
		2.0	1.000	1.000	1.000
	.25	0.8	0.995	0.996	0.997
		1.2	1.000	1.000	1.000
		1.6	1.000	1.000	1.000
		2.0	1.000	1.000	1.000
	.50	2.0	1.000	1.000	1.000

based on their proximity to both the true cluster centers and the obtained initializing values. The performance measure was $\mathcal{R}_a$ using only these classified cluster points. Scatter points were thus not included in the calculation of this performance measure, which is appropriate since the objective here is to evaluate the performance of the algorithm in obtaining the initializing cluster centers. Table 2 provides the summary measures (median, first and third quartile values) of $\mathcal{R}_a$ from the twenty-five replicates for each set of experimental scenarios. The performance of the initializing algorithm is very good, which means that it is a viable candidate for initializing our iterative algorithm.

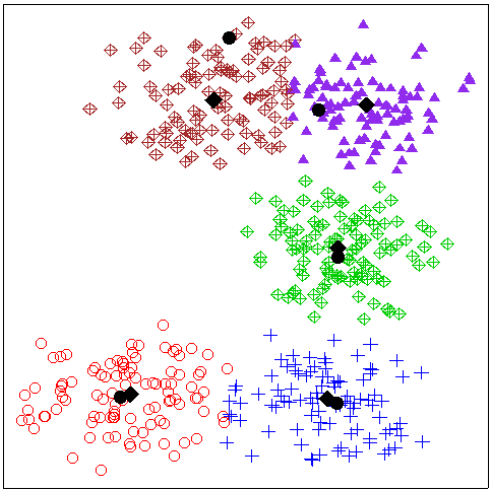
We have also experimented with using another strategy to initialization. Specifically, we have used Maitra's (2007) multi-stage algorithm for k-means after "cleaning" the dataset by removing potential scatter points, using Byers and Raftery's (1998) nearest-neighbor clutter removal technique (available as the function NNclean in the contributed R package prabclus). Table 3 shows the number of simulations where the initialization method proposed in our paper resulted in a higher, equal or lower $\mathcal{R}_a$ than that using Maitra's initializer based on the data cleaned via NNclean for 5 and 10 dimensions. The general trend shows that the initialization method proposed in the paper is best for cases of lower separation. There is a slight preference for Maitra's (2007) initialization methods after



(a) $c = 0.8$



(b) $c = 1.2$



(c) $c = 1.6$

Figure 2. Results of initialization algorithm with five clusters given as known for $c = 0.8, 1.2$,and 1.6 for the first, second and third columns respectively. Solid filled circles represent initial cluster centers, filled diamonds the true cluster centers and true cluster points are identified by both color and character.

Table 2

Summary of adjusted Rand values of groupings based on non-scattered points obtained using the initialization scheme developed for k-clips with known number of clusters K. The last three columns contain the first quartile ($\mathcal{R}_{\frac{1}{4}}$), median ($\mathcal{R}_{\frac{1}{2}}$) and third quartile ($\mathcal{R}_{\frac{3}{4}}$) of the adjusted Rand values of the twenty-five runs respectively.

	s	c	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$
$p = 5, K = 5, n = 500$	.15	0.8	0.641	0.753	0.812
		1.2	0.889	0.917	0.937
		1.6	0.958	0.972	0.989
		2.0	0.976	0.994	1.000
	.25	0.8	0.661	0.752	0.815
		1.2	0.869	0.908	0.935
		1.6	0.939	0.962	0.976
		2.0	0.985	0.994	1.000
	.50	2.0	0.979	0.992	1.000
$p = 10, K = 7, n = 2000$	.15	0.8	0.762	0.793	0.837
		1.2	0.811	0.944	0.976
		1.6	0.853	0.993	0.997
		2.0	0.846	0.997	1.000
	.25	0.8	0.720	0.771	0.822
		1.2	0.950	0.957	0.969
		1.6	0.991	0.995	0.997
		2.0	0.817	0.998	1.000
	.50	2.0	0.995	0.998	1.000
$p = 20, K = 15, n = 5000$	.15	0.8	0.859	0.877	0.883
		1.2	0.830	0.871	0.920
		1.6	0.760	0.806	0.842
		2.0	0.718	0.807	0.837
	.25	0.8	0.836	0.874	0.901
		1.2	0.819	0.923	0.996
		1.6	0.730	0.799	0.860
		2.0	0.617	0.754	0.807
	.50	2.0	0.840	0.932	1.000

NNclean when separation is high (although in many of these cases both methods resulted in essentially the same initial cluster centers). This advocates use of the proposed method or a possible hybrid, which involves running both methods and choosing the initializer optimizing the objective function as the candidate starting point.

2.3 *Evaluation of the* ***k*-clips** *algorithm with the initializer*

Table 4 provides an assessment of the performance of the k-clips algorithm together with the initializer, when K is known. Note that k-clips outperforms Mclust for the less-separated clusters. However, in most situations, as the separation increases, Mclust improves substantially relative to k-clips and eventually outperforms k-clips. Note that for $p = 2$ an equivalent conclusion is drawn in the paper.

2.4 *A Comprehensive Assessment of* ***k*-clips**

Finally, Table 5 summarizes the performance of k-clips and Mclust when the clusters need to be estimated from the data. Similar to the experiments with K known, k-clips outperforms Mclust for low separation of clusters while Mclust eventually outperforms k-clips with increasing separation between clusters. Mclust tends to under-estimate K for the lower values of c while k-clips often does very well, partially explaining the performance difference at lower

Table 3

Summary of rankings of adjusted Rand values of groupings based on non-scattered points obtained using the initialization scheme developed for k-clips and using Maitra's (2007) intializer based on data cleaned via Byers and Raftery's (1998) Nearest Neighbor Cleaning with known number of clusters K. The last three columns contain the number of times that the proposed method had the higher $\mathcal{R}$, same $\mathcal{R}$'s and lower $\mathcal{R}$ than the scheme based on the cleaned data respectively.

	s	c	High	Equal	Low
$p = 5, K = 5, n = 500$	.15	0.8	21	0	4
		1.2	21	0	4
		1.6	14	1	10
		2.0	1	9	15
	.25	0.8	22	0	3
		1.2	22	0	3
		1.6	11	0	14
		2.0	5	8	12
	.50	2	3	11	11
$p = 10, K = 7, n = 2000$	.15	0.8	25	0	0
		1.2	19	0	6
		1.6	11	0	14
		2.0	0	12	13
	.25	0.8	24	0	1
		1.2	25	0	0
		1.6	14	1	10
		2.0	2	9	14
	.50	2.0	3	11	11

Table 4

Summary of adjusted Rand values for k-clips, Mclust *and TCTW with known number of clusters K. The first quartile ($\mathcal{R}_{\frac{1}{4}}$), median ($\mathcal{R}_{\frac{1}{2}}$) and third quartile ($\mathcal{R}_{\frac{3}{4}}$) of the adjusted Rand values of the twenty-five runs is reported for each method.*

Settings			k-clips			Mclust			TCTW		
	s	c	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$
$p = 5, K = 5, n = 500$	.15	0.8	0.858	0.891	0.915	0.698	0.854	0.893	0.519	0.555	0.593
		1.2	0.952	0.973	0.984	0.884	0.951	0.970	0.655	0.676	0.749
		1.6	0.985	0.994	0.997	0.978	0.989	0.994	0.695	0.744	0.760
		2.0	0.994	0.997	1.000	0.991	0.995	1.000	0.728	0.752	0.774
	.25	0.8	0.833	0.871	0.917	0.729	0.797	0.860	0.439	0.490	0.589
		1.2	0.922	0.948	0.961	0.901	0.949	0.974	0.542	0.617	0.645
		1.6	0.959	0.970	0.977	0.964	0.981	0.993	0.632	0.673	0.705
		2.0	0.976	0.982	0.991	0.987	0.991	0.995	0.642	0.679	0.709
	.50	2.0	0.897	0.911	0.929	0.821	0.987	0.995	0.400	0.448	0.510
$p = 10, K = 7, n = 2000$	.15	0.8	0.954	0.963	0.972	0.382	0.521	0.782	0.643	0.666	0.691
		1.2	0.991	0.995	0.997	0.863	0.993	0.997	0.753	0.761	0.782
		1.6	0.998	0.999	1.000	0.853	0.999	1.000	0.782	0.802	0.810
		2.0	0.920	1.000	1.000	0.856	1.000	1.000	0.798	0.807	0.813
	.25	0.8	0.954	0.963	0.972	0.412	0.781	0.823	0.539	0.562	0.583
		1.2	0.991	0.995	0.997	0.838	0.988	0.994	0.624	0.650	0.697
		1.6	0.998	0.999	1.000	0.884	0.999	1.000	0.668	0.683	0.728
		2.0	0.920	1.000	1.000	0.866	1.000	1.000	0.661	0.675	0.744
	.50	2.0	0.998	1.000	1.000	0.999	1.000	1.000	0.383	0.422	0.443
$p = 20, K = 15, n = 5000$	.15	0.8	0.994	0.995	0.996	0.237	0.330	0.471	0.768	0.800	0.823
		1.2	0.887	0.929	0.951	0.916	0.929	0.935	0.843	0.858	0.897
		1.6	0.843	0.891	0.931	0.861	0.879	0.934	0.870	0.893	0.901
		2.0	0.833	0.887	0.918	0.866	0.923	0.932	0.852	0.866	0.904
	.25	0.8	0.993	0.994	0.995	0.196	0.394	0.534	0.645	0.678	0.708
		1.2	0.914	0.943	1.000	0.868	0.931	1.000	0.727	0.766	0.797
		1.6	0.831	0.862	0.912	0.874	0.921	0.934	0.741	0.765	0.827
		2.0	0.828	0.874	0.908	0.833	0.922	0.935	0.739	0.805	0.855
	.50	2.0	0.911	0.952	1.000	0.880	0.925	0.933	0.358	0.403	0.456

Table 5

Summary of adjusted Rand values for k-clips, Mclust *and TCTW with estimated number of clusters $\hat{K}$. The first quartile ($\mathcal{R}_{\frac{1}{4}}$), median ($\mathcal{R}_{\frac{1}{2}}$) and third quartile ($\mathcal{R}_{\frac{3}{4}}$) of the adjusted Rand values of the twenty-five runs is reported for each method.*

Settings			k-clips			Mclust			TCTW		
	s	c	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$	$\mathcal{R}_{\frac{1}{4}}$	$\mathcal{R}_{\frac{1}{2}}$	$\mathcal{R}_{\frac{3}{4}}$
$p=5, K=5, n=500$	.15	0.8	0.740	0.865	0.909	0.695	0.854	0.894	0.223	0.340	0.417
		1.2	0.943	0.971	0.984	0.934	0.951	0.966	0.479	0.555	0.635
		1.6	0.978	0.989	0.995	0.978	0.986	0.994	0.654	0.705	0.740
		2.0	0.856	0.924	0.994	0.984	0.995	1.000	0.729	0.760	0.793
	.25	0.8	0.725	0.807	0.849	0.719	0.797	0.866	0.317	0.500	0.536
		1.2	0.921	0.948	0.961	0.942	0.952	0.974	0.543	0.617	0.661
		1.6	0.958	0.970	0.977	0.967	0.984	0.994	0.653	0.742	0.801
		2.0	0.967	0.979	0.987	0.988	0.991	0.995	0.783	0.840	0.859
	.50	2.0	0.893	0.911	0.929	0.987	0.991	1.000	0.831	0.864	0.886
$p=10, K=7, n=2000$	.15	0.8	0.954	0.963	0.972	0.487	0.648	0.953	0.683	0.761	0.819
		1.2	0.906	0.994	0.996	0.990	0.995	0.997	0.863	0.916	0.975
		1.6	0.833	0.911	0.959	0.964	0.999	1.000	0.936	0.969	0.991
		2.0	0.760	0.904	0.914	0.989	0.999	1.000	0.949	0.956	0.990
	.25	0.8	0.952	0.962	0.977	0.555	0.781	0.916	0.763	0.843	0.869
		1.2	0.992	0.994	0.997	0.973	0.991	0.995	0.931	0.956	0.967
		1.6	0.856	1.000	1.000	0.991	0.999	1.000	0.978	0.981	0.985
		2.0	0.862	0.918	0.999	0.997	0.999	1.000	0.975	0.980	0.984
	.50	2.0	0.998	1.000	1.000	0.992	0.999	1.000	0.778	0.912	0.928
$p=20, K=15, n=5000$	.15	0.8	0.955	0.994	0.996	0.339	0.358	0.431	0.970	0.974	0.977
		1.2	0.756	0.858	0.917	0.974	0.985	0.996	0.980	0.982	0.984
		1.6	0.781	0.833	0.878	0.965	0.984	0.994	0.978	0.981	0.982
		2.0	0.758	0.789	0.835	0.966	0.986	0.995	0.979	0.981	0.982
	.25	0.8	0.993	0.994	0.995	0.329	0.407	0.472	0.925	0.940	0.945
		1.2	0.841	0.960	0.999	0.963	0.983	1.000	0.951	0.960	0.969
		1.6	0.840	0.880	0.932	0.966	0.979	0.994	0.956	0.962	0.966
		2.0	0.736	0.820	0.866	0.960	0.982	0.989	0.924	0.951	0.957
	.50	2.0	0.934	1.000	1.000	0.986	0.995	1.000	0.535	0.617	0.884

separations. As seen in Table 6, for higher values of c, in particular when $p = 20$, k-clips tends to under-estimate K. This tendency is not as pronounced for $p = 10$ and is essentially non-existent for $p = 5$. In the situations where k-clips over-estimates K, it tends to create several smaller clusters of what is actually scatter while (essentially) correctly identifying the true clusters. The tendency to overestimate K is observed in other methods too, very notably in Mclust and nearly universally in TCTW. For k-clips, this tendency is only apparent when $p = 20$.

From the series of evaluations provided, it appears that both k-clips and Mclust have strengths and weaknesses. While k-clips provides better groupings than Mclust in datasets with low to moderate separation of clusters, Mclust provides better results for well-separated clusters. Of course, it is worth noting that the model generating the observations is the likelihood model on which Mclust is explicitly based.

2.5 *Evaluating Sensitivity of* ***k*-clips** *to Gaussian Assumptions*

The k-clips algorithm was motivated using an underlying hard-clustering spherical-Gaussians model. To assess sensitivity of model assumption violations on performance, we also evaluated performance on a series of two-dimensional non-Gaussian settings. These scenarios included generating datasets from mixtures of spherical but heavy-tailed distributions, irregular crescent-shaped and variedly-oriented clusters, data constrained to lie on a circle or clusters highly skewed in both dimensions. Two datasets, each containing 100 scatter and 500 observations from amongst five clusters with equal mixing proportions were generated in each case. One dataset had clusters with low-separation the other with high separation. As before, we compared the performance of k-clips, Mclust and TCTW.

2.5.1 *Heavy-tailed Clusters.* Observations from the heavy-tailed clusters were simulated by generating realizations from independent t-distributions with 3 degrees of freedom in each coordinate and then adding to the cluster means following a similar procedure used in the c-separated Gaussians in the main experiments.

Figure 3 shows that the results for the three methods with K known for both the low and high separated cases are very similar to the Gaussian case. It is encouraging to note that k-clips is remarkably robust to heavy tailed spherical distributions with only a few scattered points being included in the clusters. Thus, we see that choosing robust scale estimation not only does a good job of identifying cores, but also in identifying spherical but heavy-tailed clusters from scatter. Note that Mclust, while still doing well, has a tendency to misclassify the outskirts of the clusters as scatter, a result of the strict Gaussian assumptions inbuilt in modeling the clusters. TCTW again exhibits problems similar to those seen in the Gaussian case, the most severe being the failure to adhere to

Table 6

Summary of the estimated number of clusters $\hat{K}$ for k-clips, Mclust *and TCTW. The first quartile ($\hat{K}_{\frac{1}{4}}$), median ($\hat{K}_{\frac{1}{2}}$) and third quartile ($\hat{K}_{\frac{3}{4}}$) of $\hat{K}$ of the twenty-five runs is reported for each method.*

Settings			k-clips			Mclust			TCTW		
	s	c	$\hat{K}_{\frac{1}{4}}$	$\hat{K}_{\frac{1}{2}}$	$\hat{K}_{\frac{3}{4}}$	$\hat{K}_{\frac{1}{4}}$	$\hat{K}_{\frac{1}{2}}$	$\hat{K}_{\frac{3}{4}}$	$\hat{K}_{\frac{1}{4}}$	$\hat{K}_{\frac{1}{2}}$	$\hat{K}_{\frac{3}{4}}$
$p=5, K=5, n=500$	.15	0.8	5	5	5	4	5	5	14	15	15
		1.2	5	5	5	5	5	6	14	15	15
		1.6	5	5	5	5	5	5	15	15	15
		2.0	4	5	5	5	5	5	15	15	15
	.25	0.8	4	5	5	4	5	5	15	15	15
		1.2	5	5	5	5	5	6	15	15	15
		1.6	5	5	5	5	5	5	15	15	15
		2.0	5	5	5	5	5	5	15	15	15
	.50	2.0	5	5	5	5	5	6	15	15	15
$p=10, K=7, n=2000$	.15	0.8	7	7	7	4	7	8	13	14	18
		1.2	7	7	7	7	7	8	16	17	18
		1.6	6	6	7	7	8	8	16	17	19
		2.0	5	6	6	7	8	9	17	18	19
	.25	0.8	7	7	7	5	6	8	15	16	17
		1.2	7	7	7	7	8	8	15	16	17
		1.6	6	7	7	7	8	8	15	17	17
		2.0	6	6	7	7	7	8	16	18	19
	.50	2.0	7	7	7	7	8	8	10	11	13
$p=20, K=15, n=5000$	.15	0.8	15	15	15	4	5	5	21	22	23
		1.2	11	13	13	16	17	18	20	21	22
		1.6	11	12	14	16	17	18	19	20	22
		2.0	10	11	11	16	17	19	19	21	22
	.25	0.8	15	15	15	4	5	6	16	17	17
		1.2	15	15	16	15	16	18	16	17	18
		1.6	14	15	17	16	18	19	15	16	17
		2.0	12	13	15	16	17	18	15	16	18
	.50	2.0	15	15	16	16	17	18	15	19	20

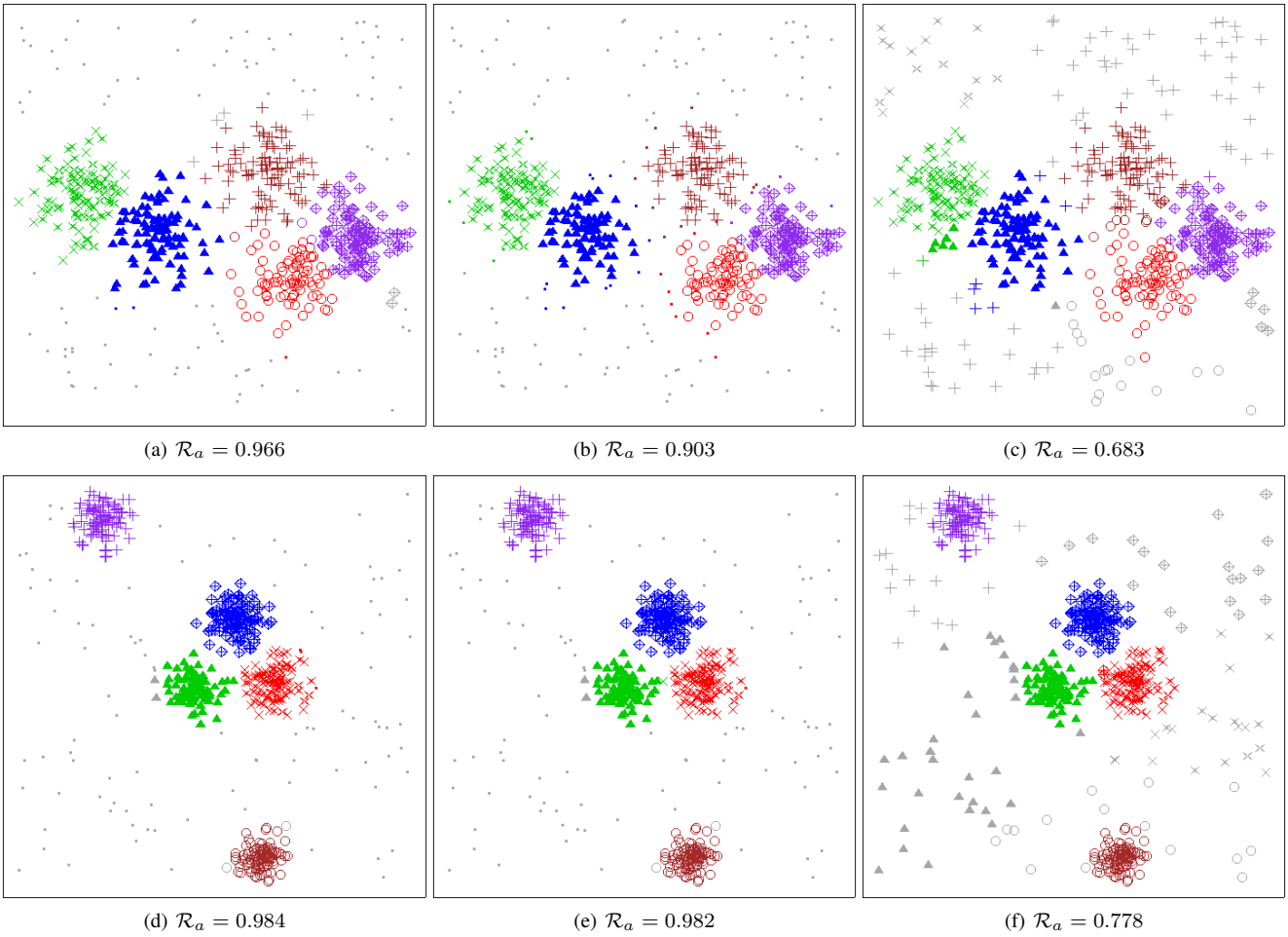
the distance metric. Finally, when estimating the optimal K, both k-clips and Mclust correctly identify five clusters for both datasets, indicating that the Gaussian assumption may also not be that crucial in estimating the number of spherical heavy-tailed clusters. Note that TCTW again chooses all fifteen target clusters as optimal, with $\mathcal{R}_a = 0.47$ for the moderately-separated clusters and $\mathcal{R}_a = 0.609$ for the well-separated clusters.

2.5.2 *Irregularly-shaped Clusters.* We also assessed the performance of k-clips on irregularly shaped clusters to understand its performance when clusters deviate substantially from the inherent spherical cluster assumption. In this experiment, we generated clusters from c-separated bivariate standard normal density outside of a circle of a randomly chosen radius between 0 and 3. Each cluster was then "folded" over a randomly chosen bisector resulting in "half-ring" clusters with different orientations.

Figure 4 shows that with low separation and known K, all three methods correctly identify portions of each cluster with k-clips and Mclust performing similarly. All three methods also identify the cluster edges incorrectly with TCTW having the most difficulty. For the corresponding well-separated case, all methods performed better with k-clips besting the others. Mclust has the most problems as it combines two clusters together while classifying a group of scatter as an actual cluster. TCTW correctly finds the clusters, but tended to include a large amount of scatter within each identified cluster.

For unknown K for both the low and high separation cases, k-clips overestimated the true number of clusters by splitting the true clusters into several smaller "regular-shaped" ones (Figure 5). Additionally in the low separation case, k-clips misclassified a large portion of one cluster as scatter. Mclust split the true clusters into numerous smaller clusters for the low separation case but did surprisingly well ($\mathcal{R}_a = 0.994$) in the high separation case: although it did combine a small group of scatter into a new cluster. TCTW tended to split apart existing clusters and combined groups of scatter into new clusters. Thus, we note that performance of k-clips can be severely affected when clusters are widely irregular and K is unknown. The algorithm does remain robust with known K, especially in the case of well-separated clusters.

2.5.3 *Clustering Directional Data.* Many clustering algorithms are applied to data from microarray experiments, for example, in clustering gene expression profiles (Dortet-Bernadet and Wicker, 2008). It is often the case that the researcher is interested in determining groups based on the correlation between these profiles. In such cases, a common strategy is to standardize each observation to have zero mean and unit variance. This transformation essentially



(a) $\mathcal{R}_a = 0.966$ (b) $\mathcal{R}_a = 0.903$ (c) $\mathcal{R}_a = 0.683$

(d) $\mathcal{R}_a = 0.984$ (e) $\mathcal{R}_a = 0.982$ (f) $\mathcal{R}_a = 0.778$

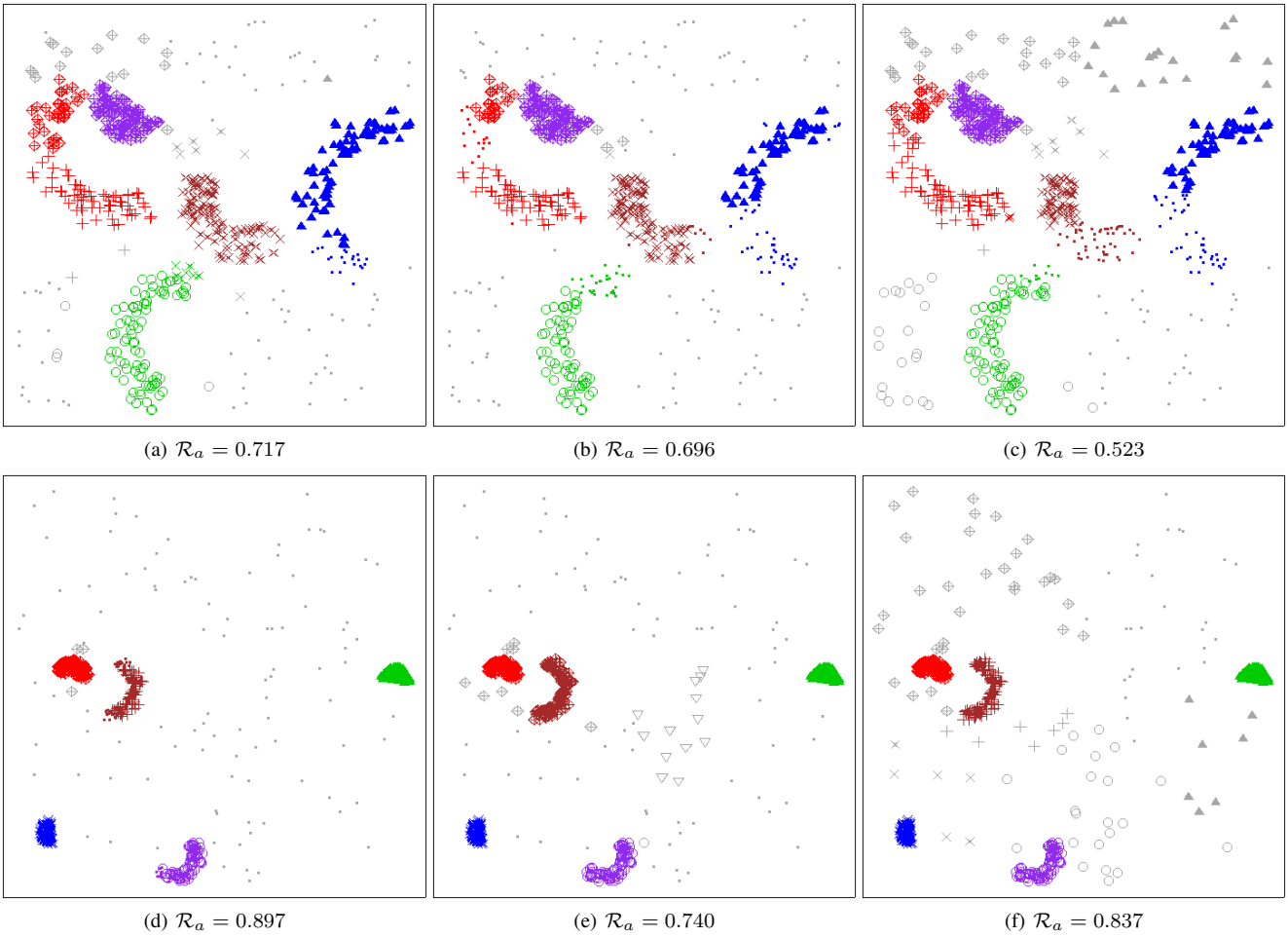
Figure 3. Classifications of clusters generated from independent marginal t-distributions with 3 degrees of freedom based on k-clips (first column), Mclust (second column) and TCTW (third) with five clusters given as known for low separation (top row) and high separation (bottom row). Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.

places observations on a unit sphere, but constrained to be orthogonal to the unit vector. The first $p-1$ principle components then lie unconstrained on an unit sphere of $p-1$ dimensions. Thus, in a three-dimensional dataset, there is no loss in projecting the standardized dataset onto the circumference of a unit circle. To assess the performance of the algorithm in these conditions, clusters were generated from a mixture of von-Mises distributions with moderate to high concentrations (reflecting cases of low and high separation) and scatter was added uniformly around the circumference of the circle.

Figure 6 shows that for known K, k-clips does an excellent job in distinguishing the moderately-concentrated clusters and scatter ($\mathcal{R}_a = 0.991$). Mclust included a large amount of scatter in each cluster resulting in a lower $\mathcal{R}_a = 0.662$ while TCTW did slightly better, even though it failed to identify one of the true clusters ($\mathcal{R}_a = 0.714$). For the more concentrated clusters, k-clips consistently underestimated the size of the cores while Mclust correctly identified most of the observations ($\mathcal{R}_a = 0.988$). Mclust's improved performance was not unexpected as for high concentrations the von-Mises distribution can be adequately approximated by an univariate Gaussian distribution (Mardia and Jupp; 2000). Mclust is general enough to adapt to this lower dimensional Gaussian situation by choosing the dispersion matrix appropriately, unlike k-clips which did not use this additional information and so still found clusters in two dimensions. Further, it used robust estimates of scale and consistently underestimated the cores. TCTW showed improved performance with higher separation, even though it included large amounts of scatter in each cluster.

As seen in Figure 7, when K is unknown, for both datasets, k-clips underestimated the true number of clusters by considering several of the true clusters as scatter. Conversely, Mclust grossly overestimated the number of clusters by continually splitting clusters and combining groups of scatter into clusters. This characteristic was much more severe in the moderately concentrated case as $\hat{K} = 28$ than in the high separation case where $\hat{K} = 14$. TCTW also tended to both split clusters and combine scatter in groups for each case, resulting in low values of $\mathcal{R}_a$.

2.5.4 *Clusters embedded within highly skewed datasets.* As seen in the mercury release application in the main paper, clustering highly skewed datasets can be a difficult task. To assess the performance of the algorithms in such situations, clusters were generated



(a) $\mathcal{R}_a = 0.717$ (b) $\mathcal{R}_a = 0.696$ (c) $\mathcal{R}_a = 0.523$

(d) $\mathcal{R}_a = 0.897$ (e) $\mathcal{R}_a = 0.740$ (f) $\mathcal{R}_a = 0.837$

Figure 4. Classifications of irregularly shaped clusters based on k-clips (first column), Mclust (second column) and TCTW (third) with five clusters given as known for low separation (top row) and high separation (bottom row). Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.

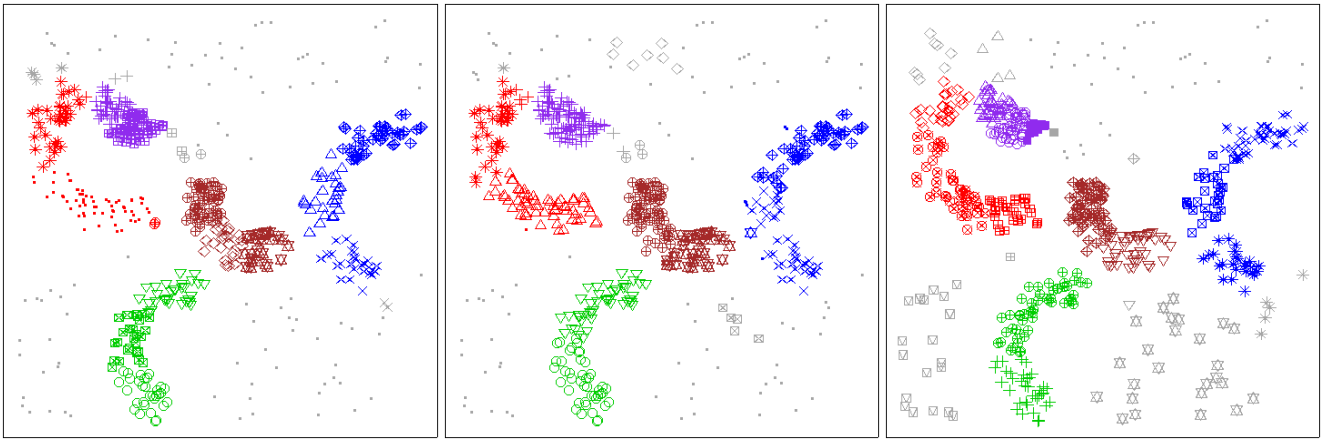
using the c-separated Gaussian method described in Section 3 of the main paper. Each dataset was then transformed to follow a log-log distribution to incorporate high levels of skewness.

Figure 8 shows that with K known for the low separation case, all three methods perform very poorly (Mclust performed "best" with $\mathcal{R}_a = 0.443$). For the highly separated clusters, performance still lacked quality. k-clips did the best ($\mathcal{R}_a = 0.723$) as it correctly identified most of the scatter, but failed to identify large portions of true clusters. Mclust did the opposite by misclassifying most of the scatter while correctly identifying more of the clustered points. TCTW did find some of the clusters but included large amounts of scatter as part of each cluster.

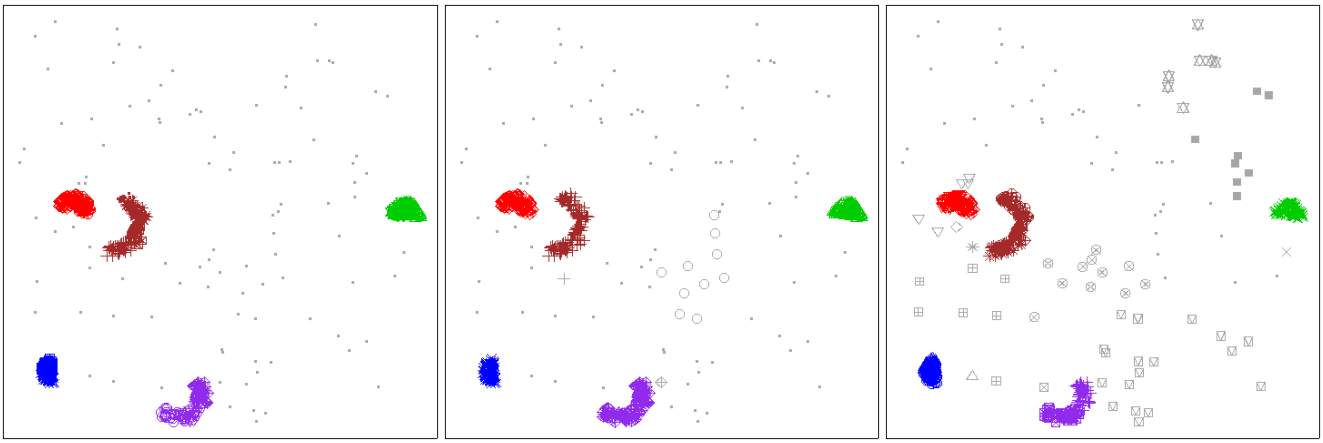
When the optimal K was estimated (Figure 9), k-clips underestimated the number of clusters for the low separation case by combining several clusters while Mclust slightly overestimated K, resulting in slightly better classifications than when K was given. TCTW split the dataset into numerous clusters but also failed to find the true underlying clusters. For the high separation, k-clips still found five optimal clusters (albeit not the five true clusters) while Mclust again slightly overestimated K. TCTW does somewhat better at identifying clusters (while still splitting some the true clusters apart), but classified the vast majority of the scatter into clusters as well.

When the data is log-log transformed (Figure 10) performance improves dramatically for all three methods. For the low separation case with K known, the classification provided by$k$-clips improves to $\mathcal{R}_a = 0.757$ and Mclust's to $\mathcal{R}_a = 0.759$. Due to the low separation between some of the clusters, k-clips combines two clusters together (and finds a small cluster of scatter) while Mclust has difficulties determining the borders of each cluster. TCTW improves as well, but again misclassifies the majority of the scatter and identifies a large portion of one cluster as scatter. When K is unknown, Mclust finds the same five clusters as before. k-clips finds four optimal clusters, discarding the small cluster of scatter identified when K was given. TCTW finds all 15 target clusters but still misclassifies a portion of a cluster as scatter. For the high separation case both k-clips and Mclust are nearly perfect with $\mathcal{R}_a$ near one. TCTW again adds nearly all of the scatter into the nearest cluster. When K is unknown both k-clips and Mclust correctly identify five clusters while TCTW finds all 15 target number of clusters as directed.

As seen in these examples, k-clips still performs well in many situations outside of Gaussian clusters. In particular it seems to



(a) $\mathcal{R}_a = 0.464$, $\hat{K} = 12$ (b) $\mathcal{R}_a = 0.688$, $\hat{K} = 11$ (c) $\mathcal{R}_a = 0.539$, $\hat{K} = 15$



(d) $\mathcal{R}_a = 0.544$, $\hat{K} = 12$ (e) $\mathcal{R}_a = 0.994$, $\hat{K} = 6$ (f) $\mathcal{R}_a = 0.664$, $\hat{K} = 15$

Figure 5. Classifications of irregularly shaped clusters based on k-clips (first column), Mclust (second column) and TCTW (third) with the number of clusters estimated for low separation (top row) and high separation (bottom row). Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.

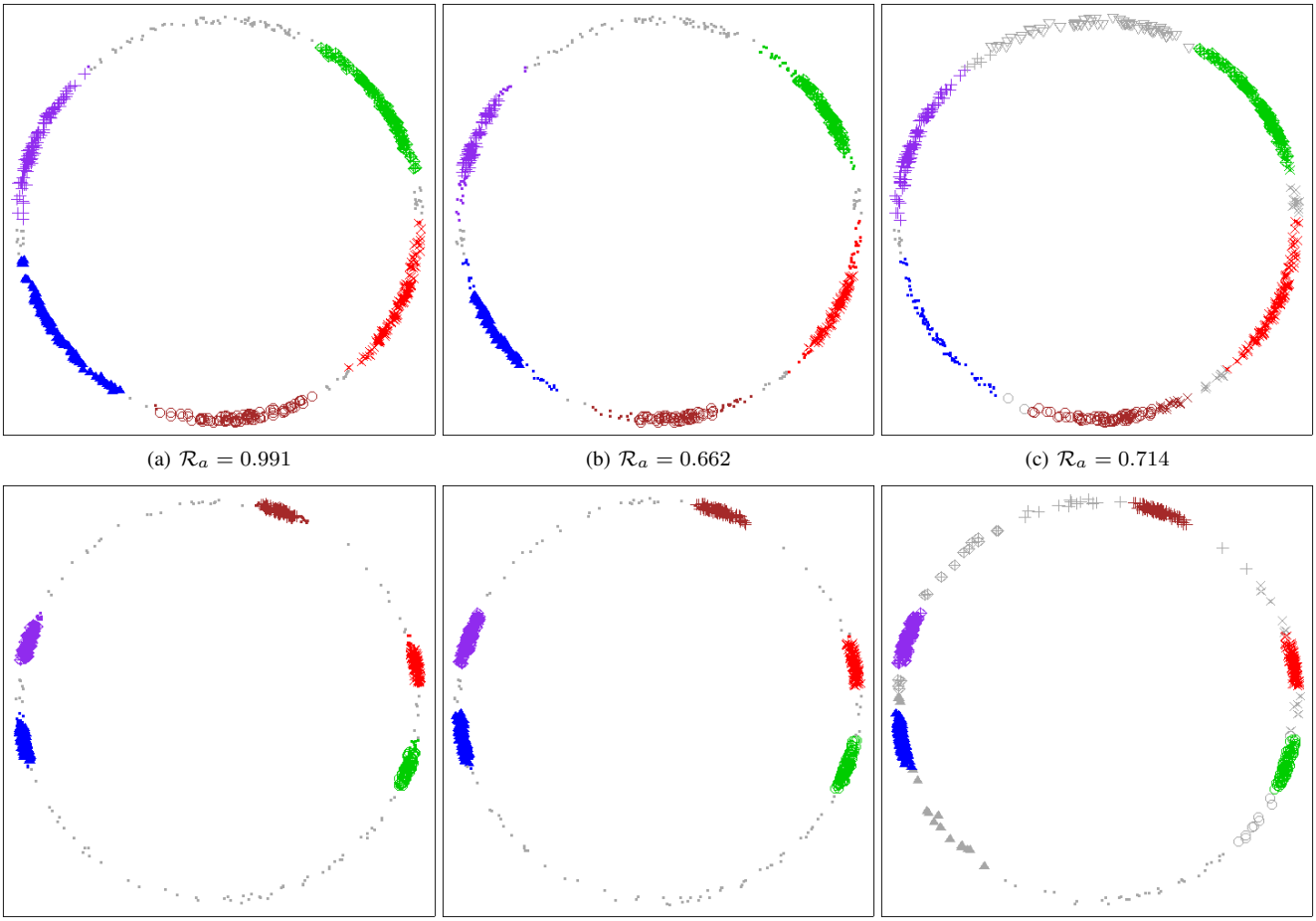
be the most robust to heavy tailed clusters like those generated from a mixture of independent t-distributions. These are cases when the clusters are still spherical in shape. k-clips however performed poorly when the data (and underlying clusters) are highly skewed as in the log-log distribution case. In such cases, transforming the dataset improves performance appreciably. It also failed to correctly identify the number of clusters for datasets with widely irregular clusters or constrained datasets, such as in the directional case. Of course, neither Mclust or TCTW performed well in many of these situations also.

References

Dortet-Bernadet, J-L and Wicker, N., (2008). Model-based clustering on the unit sphere with an illustration using gene expression profiles. *Biostatistics* 2008 9(1):66-80.

Mardia, K. V. and Jupp, P. E., (2000). *Directional Statistics* (2nd edition). John Wiley and Sons, Inc. NY.

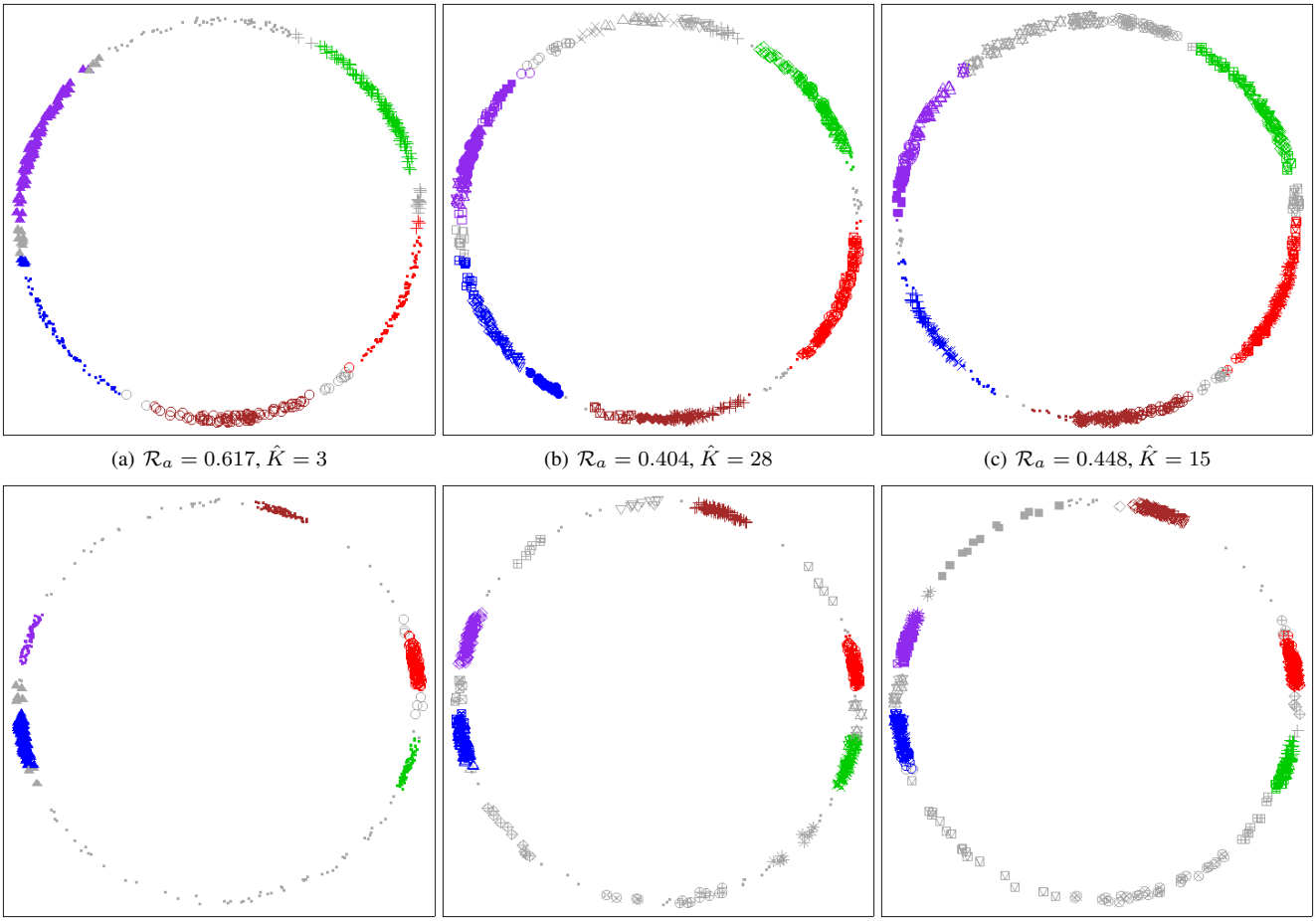
Received December 2006. *Revised June* 2007.
Accepted March 2008.



(a) $\mathcal{R}_a = 0.991$ (b) $\mathcal{R}_a = 0.662$ (c) $\mathcal{R}_a = 0.714$

(d) $\mathcal{R}_a = 0.767$ (e) $\mathcal{R}_a = 0.988$ (f) $\mathcal{R}_a = 0.846$

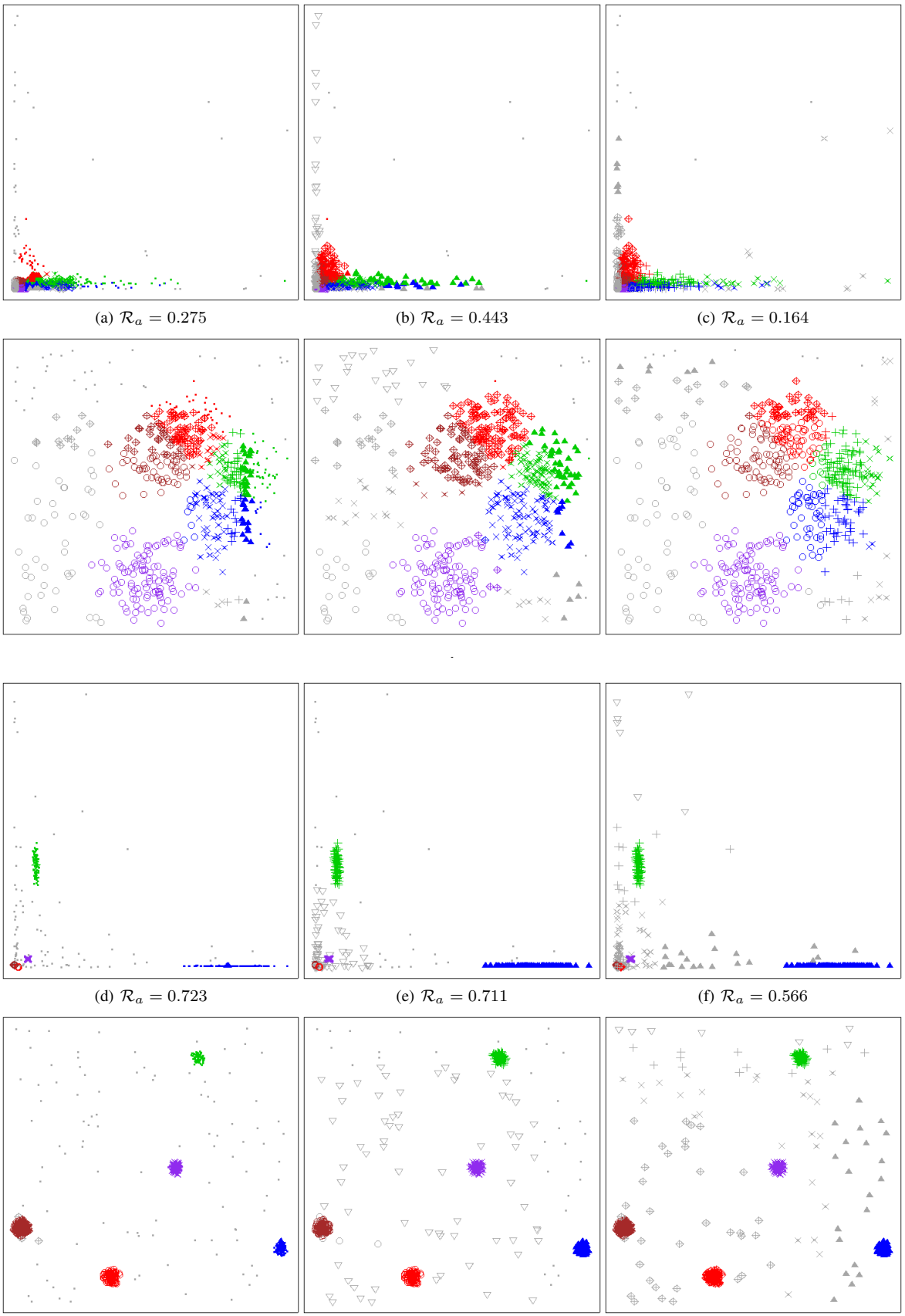
Figure 6. Classifications of von-Mises clusters based on k-clips (first column), Mclust (second column) and TCTW (third) with five clusters given as known for low separation (top row) and high separation (bottom row). Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.



(a) $\mathcal{R}_a = 0.617$, $\hat{K} = 3$ (b) $\mathcal{R}_a = 0.404$, $\hat{K} = 28$ (c) $\mathcal{R}_a = 0.448$, $\hat{K} = 15$

(d) $\mathcal{R}_a = 0.558$, $\hat{K} = 2$ (e) $\mathcal{R}_a = 0.947$, $\hat{K} = 14$ (f) $\mathcal{R}_a = 0.559$, $\hat{K} = 15$

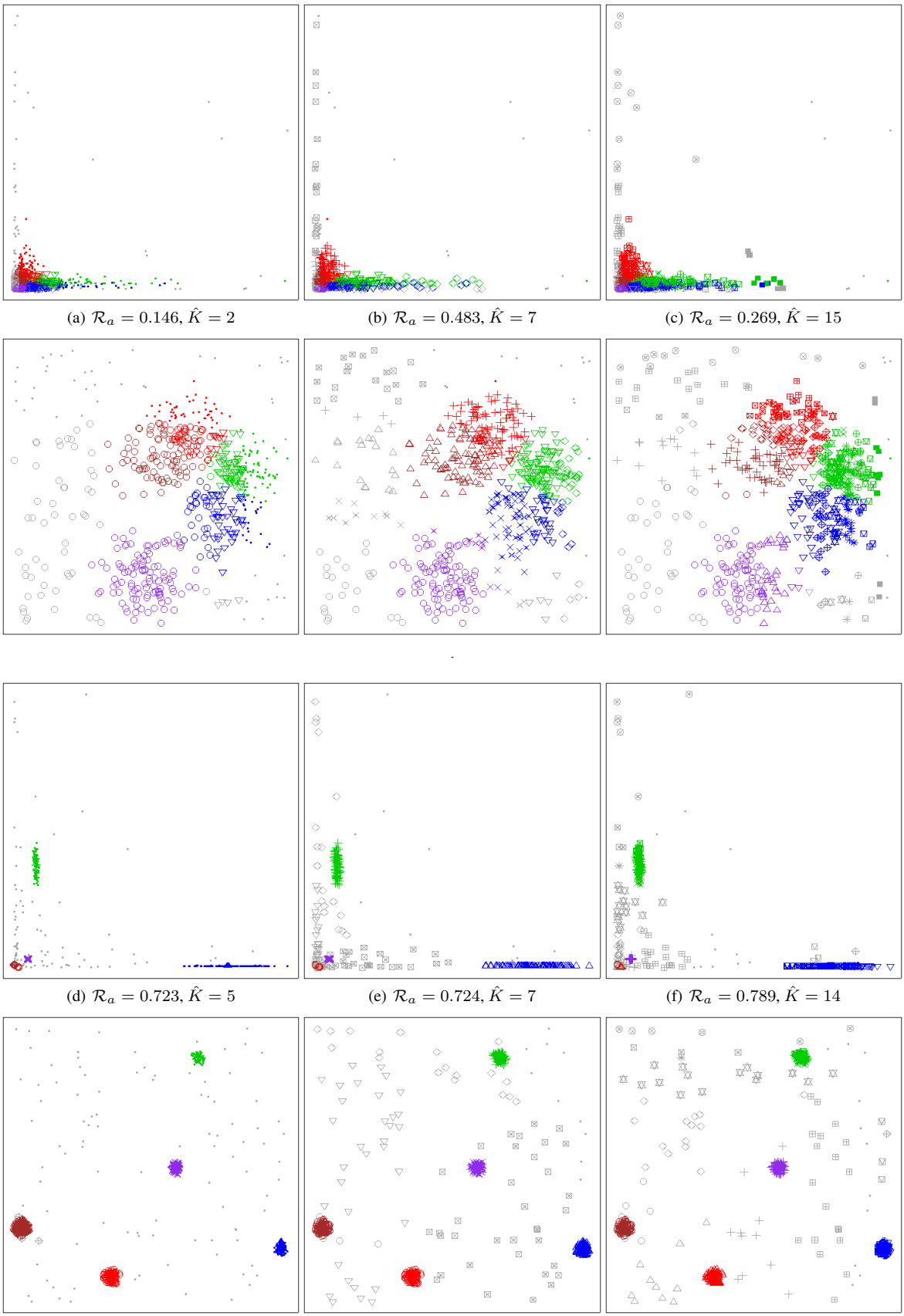
Figure 7. Classifications of von-Mises clusters based on k-clips (first column), Mclust (second column) and TCTW (third column) with the number of clusters estimated for low separation (top row) and high separation (bottom row). Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.



(a) $\mathcal{R}_a = 0.275$ (b) $\mathcal{R}_a = 0.443$ (c) $\mathcal{R}_a = 0.164$

(d) $\mathcal{R}_a = 0.723$ (e) $\mathcal{R}_a = 0.711$ (f) $\mathcal{R}_a = 0.566$

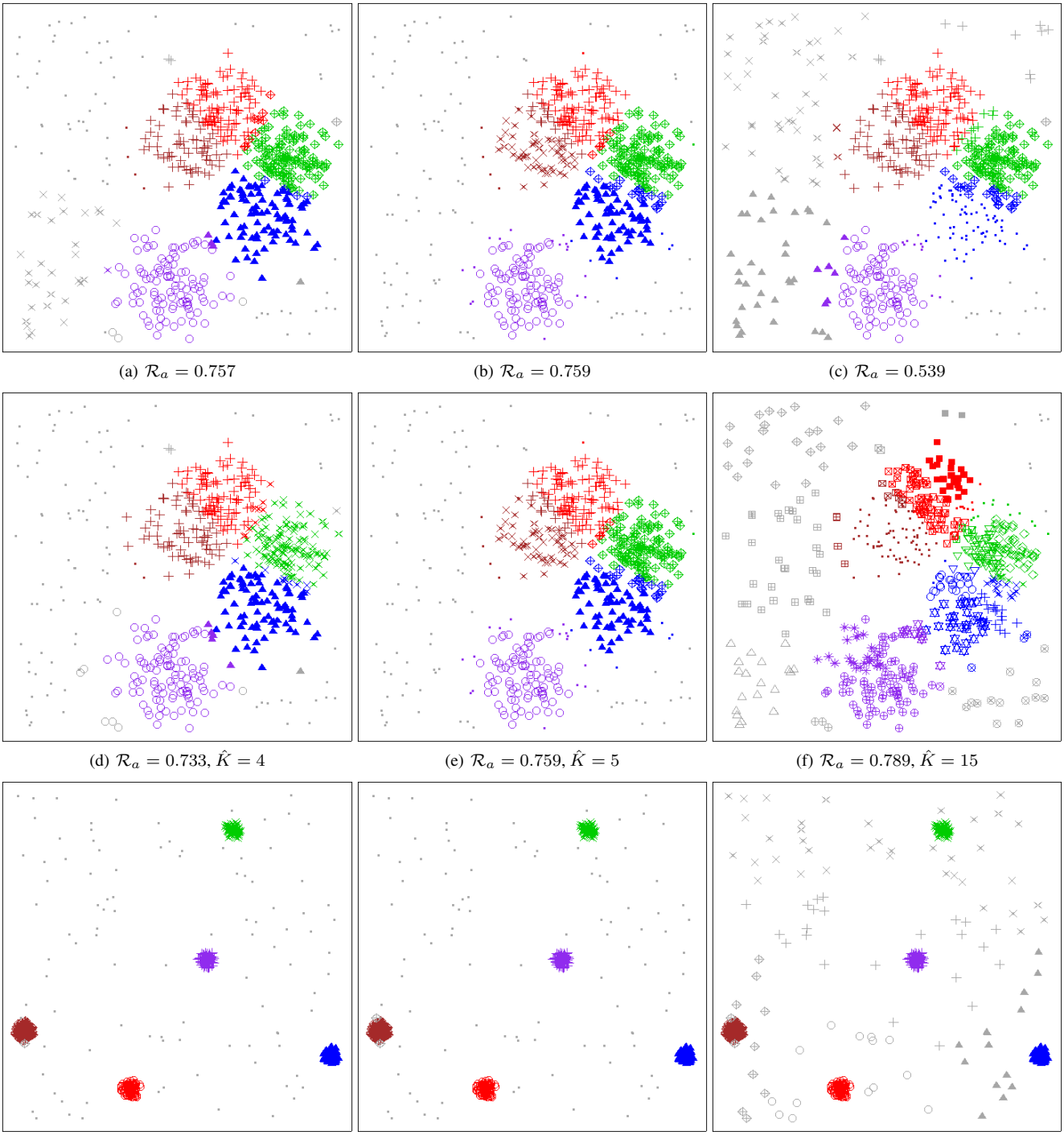
Figure 8. Classifications of highly skewed clusters based on k-clips (first column), Mclust (second column) and TCTW (third column) with five clusters given as known for low separation (top two rows) and high separation (bottom two rows) where the second and fourth rows are shown in log-log scale. Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.



(a) $\mathcal{R}_a = 0.146$, $\hat{K} = 2$ (b) $\mathcal{R}_a = 0.483$, $\hat{K} = 7$ (c) $\mathcal{R}_a = 0.269$, $\hat{K} = 15$

(d) $\mathcal{R}_a = 0.723$, $\hat{K} = 5$ (e) $\mathcal{R}_a = 0.724$, $\hat{K} = 7$ (f) $\mathcal{R}_a = 0.789$, $\hat{K} = 14$

Figure 9. Classifications of highly skewed clusters based on k-clips (first column), Mclust (second column) and TCTW (third column) with the number of clusters estimated for low separation (top two rows) and high separation (bottom two rows) where the second and fourth rows are shown in log-log scale. Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.



(a) $\mathcal{R}_a = 0.757$ (b) $\mathcal{R}_a = 0.759$ (c) $\mathcal{R}_a = 0.539$

(d) $\mathcal{R}_a = 0.733$, $\hat{K} = 4$ (e) $\mathcal{R}_a = 0.759$, $\hat{K} = 5$ (f) $\mathcal{R}_a = 0.789$, $\hat{K} = 15$

(g) $\mathcal{R}_a = 0.997$ (h) $\mathcal{R}_a = 0.995$ (i) $\mathcal{R}_a = 0.793$

Figure 10. Classifications of highly skewed data transformed to log-log scale based on k-clips (first column), Mclust (second column) and TCTW (third column) with the number of clusters given for low separation (top row) and high separation (third row). Results when number of clusters is estimated for low separation given in second row. Small filled circles represent identified scatter, colors - true clusters and characters - identified clusters.