Multivariate permutation tests for the k-sample problem with clustered data [1]

Jörg Rahnenführer[2]

[2] Department of Statistics, Vienna University of Economics and Business Administration, Augasse 2-6, A-1090 Vienna, Austria

Abstract:

The present paper deals with the choice of clustering algorithms before treating a k-sample problem. We investigate multivariate data sets that are quantized by algorithms that define partitions by maximal support planes (MSP) of a convex function. These algorithms belong to a wide class containing as special cases both the well known k-means algorithm and the Kohonen (1985) algorithm and have been profoundly investigated by Pötzelberger and Strasser (1999). For computing the test statistics for the k-sample problem we replace the data points by their conditional expections with respect to the MSP-partition. We present Monte Carlo simulations of power functions of different tests for the k-sample problem whereas the tests are carried out as multivariate permutation tests to ensure that they hold the level. The results presented show that there seems to be a vital and decisive connection between the optimal choice of the clustering algorithm and the tails of the probability distribution of the data. Especially for distributions with heavy tails like the exponential distribution the performance of tests based on a quadratic convex function with k-means type partitions totally breaks down.

Keywords: Data compression, Clustering, MSP-partitions, Multivariate Permutation tests, k-sample problem

[1] This work was supported by the Spezialforschungsbereich 010 "Adaptive Information Systems and Modelling in Economics and Management Science", Vienna University of Economics and Business Administration, funded by the Austrian Fonds zur Förderung der wissenschaftlichen Forschung.

1 Introduction

Especially in multivariate test problems the selection of a special class of models often cannot be justified because a high number of parameters had to be adjusted. For the estimation of the unknown parameters one needs in turn a large number of observations rapidly growing with the dimension. One possibility to overcome this so-called 'curse of dimensionality' is to first apply a clustering algorithm to the data and then adopt a model class to the new data set represented by some prototypes. The first step in this two stage procedure can be seen as an initialization phase to be able to reduce the model complexity afterwards. In this paper we investigate the influence of different quantization or clustering partitions if the underlying test problem is a k-sample problem. Let us first turn to the description of the phase of data reduction.

Statistical cluster analysis is concerned with the classification or grouping of data. In this article we deal with the reduction of data sets to partitions $\mathcal{B} = (B_1, \ldots, B_m)$, i.e. to m disjoint groups. The quality of such a partition can be measured by the f-Information which is defined for arbitrary measures P on $\mathbb{R}^d$ with existing first moment as follows.

Definition 1. *(f-Information)*
Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function. The f-information of $\mathcal{B} = (B_1, \ldots, B_m)$ (with respect to the distribution P) is given by

$$I_f(\mathcal{B}) := \sum_{j=1}^{m} P(B_j)\, f(m(B_j)), \tag{1}$$

where for every set B_j in the partition the mean is given by

$$m(B_j) := \frac{1}{P(B_j)} \int_{B_j} x\, P(dx), \qquad \text{if } P(B_j) > 0. \tag{2}$$

One importanrt reason of the restriction to convex functions ensures that a division of one set of the partition into two new ones cannot lead to a not at all plausible lower f-Information. The goal of the clustering method then is to maximize this functional.

Definition 2. *(Optimization problem)*
Find a partition $\mathcal{B} = (B_1, \ldots B_m)$ such that

$$I_f(\mathcal{B}) \;=\; \texttt{Max!} \tag{3}$$

under $|\mathcal{B}| \leq m$.

Let us focus on two special cases. For the choice $f(x) = |x|^2$ the equation

$$I_f(\mathcal{B}) \;=\; \int |x|^2 \, P(dx) - \sum_{j=1}^{m} \int_{B_j} |x - m(B_j)|^2 \, P(dx) \tag{4}$$

shows that a maximization of $I_f(\mathcal{B})$ is equivalent to a minimization of the the inner dispersion of the partition $\mathcal{B}$, wherefore such a solution is also called a minimum variance partition. In this case the most popular algorithm to solve the optimization problem 3 is the k-means algorithm. It improves a starting solution by alternating between computing centers of a given partition and constructing a new partition by assigning every point in $\mathbb{R}^d$ to its nearest neighbour in L_2-distance of these centers. Detailed and advanced results for this special case can be found e.g. in Bock (1964).

The second outstanding case is the function $f(x) = |x|$ that leads to the Kohonen algorithm (Kohonen, 1984). Then the resulting partition is independent of the distance of the data points from the origin 0, therefore outliers have no influence on the result.

For the general case of an arbitrary convex function a unifying theory has been developed by Bock (1992) for the one dimensional case and in higher dimensions by Pötzelberger and Strasser (1999), who define a fixpoint algorithm which is a slight modification in the two special cases mentioned above. But the choice of a general convex function aditionally offers a variety of new clustering procedures. In chapter 2 we will briefly specify the precise algorithm and explain its usage by examples.

Since we are interested in statistical inference, we compare the quality of the different algorithms with the help of simulation experiments. To obtain criteria for the application of the specific convex functions we use the multivariate k-sample problem which is a comparison of k samples drawn from the same multivariate distribution and differing only in respect of a location parameter. Particularly we first apply the fixpoint algorithm to the pooled sample to reduce the data set to some clusters represented by prototypes and then replace the original data points by their prototypes, respectivly.

To find differences between the k samples two types of test statistics are taken into account. The first one measures the inner variance of the clustered data set and the second one is a χ^2-type statistic. The exact formulation of the test problem and the test statistics are given in chapter 3. There we also describe the determination of the critical values based on the evaluation of the tests as permutation tests.

In chapter 4 we state explicitely the parameter constellations tested and present the results of the simulation experiments for the power functions, in chapter 5 we discuss these results and draw conclusions about the appropriate choice of a clustering algorithm depending on the properties of underlying

distributions.

2 Clustering of data by MSP-Partitions

In this section we present the fixpoint algorithm of Pötzelberger and Strasser (1999) which is a used in our simulations to compress the data sets and which generalizes the well known k-means algorithm. The fixpoint algorithm finds local optima of the optimization problem (3) for an arbitrary but fixed convex function f whereas k-means is applied to the special case $f(x) = |x|^2/2$. The alternation in k-means between constructing a new partition based on some prototypes and computing the centers of this partition as new prototypes is kept, but the partitioning step is more complicate. To present the idea and state the exact algorithm we first give some definitions and notions.

Definition 3. *(Conjugate convex function)*
Let $f : \mathbb{R}^d \to \mathbb{R}$ be a convex function. Then

$$f^c(a) := \sup_x (a'x - f(x)), \quad a \in \mathbb{R}^d \tag{5}$$

is called the conjugate convex function of f.
The support of the conjugate convex function is denoted by

$$A := K(f^c) = \{x \in \mathbb{R}^d : f_c(x) < \infty\}. \tag{6}$$

From convex analysis it is known that $f_c(a)$ is maximal such that $l(x) := a'x - f_c(a) \leq f(x)$ holds. The affine linear function l is called *maximal support function* of f in x with slope a. Such functions are the tool to a general construction of partitions which are called MSP-partitions, definition (4.9) in Pötzelberger and Strasser (1999).

Definition 4. *(MSP-partition)*
A partition $\mathcal{B} = (B_1, B_2, \ldots, B_m)$ is called a MSP-partition (maximum support plane partition) if there exists $\mathbf{a} \in A^m$ such that for $j = 1, 2, \ldots, m$:

$$x \in B_j \quad \Rightarrow \quad a_j'x - f_c(a_j) = \max_{1 \leq k \leq m} \left(a_k'x - f_c(a_k)\right). \tag{7}$$

Let $\mathcal{S}(\mathbf{a})$ be the family of all MSP-partitions associated with $\mathbf{a}$ and let $\mathcal{S}_m = \bigcup_{\mathbf{a} \in A^m} \mathcal{S}(\mathbf{a})$.

This means that MSP-partitions are constructed by maximal support planes of the convex function under consideration. For given slopes a_j, $j = 1, \ldots m$ each point is assigned to the partition for which the value of the appropriate support plane is maximal.

Figure 1 demonstrates this procedure in the one dimensional case for the functions $f(x) = |x|^2/2$ and $f(x) = |x|^{1.5}/1.5$. The maximal support functions are plotted at the points -1.5, -0.5 and 2, and their intersections are the borders of the resulting partition.

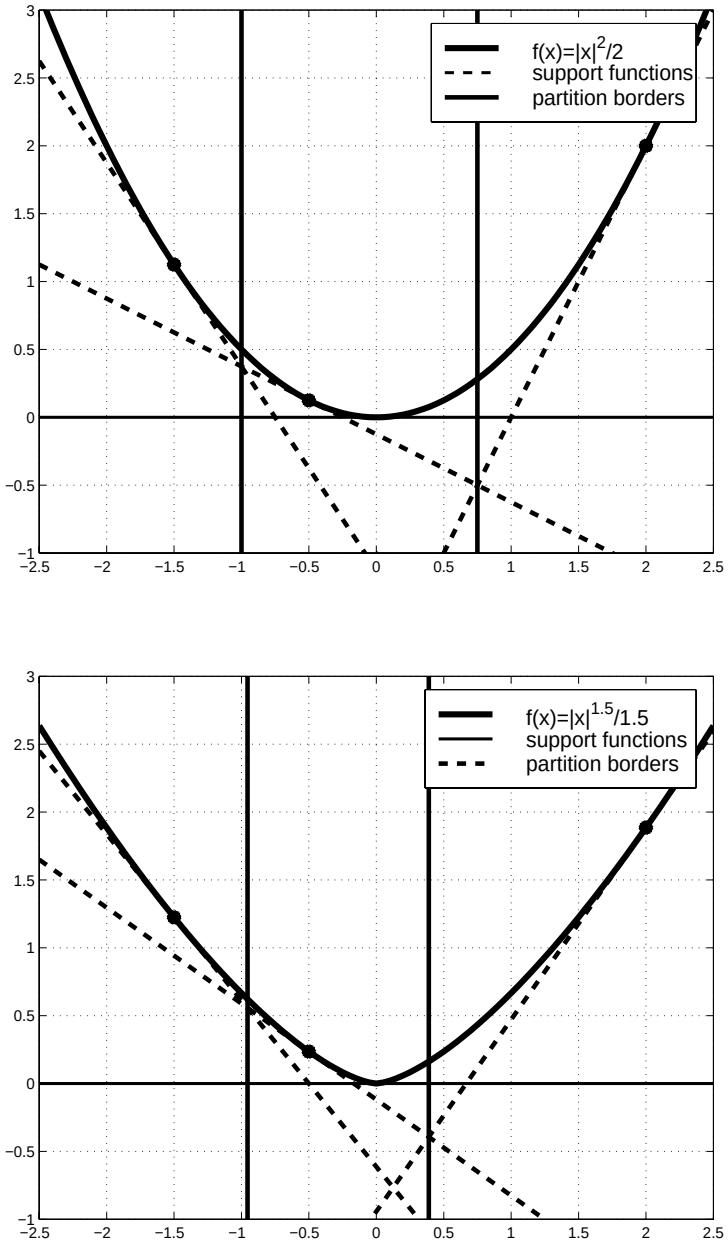

Figure 1: MSP-partitions for $f(x) = |x|^2/2$ (left) and $f(x) = |x|^{1.5}/1.5$ (right)

We see that the application of $f(x) = |x|^{1.5}/1.5$ instead of $f(x) = |x|^2/2$ leads to a more directional view as the partition belonging to the only positive pro-

totype is larger. The most extreme case $f(x) = |x|$ leads to a segmentation with just two partitions separated at $x = 0$, a purely directional based classification.

With the above preparations the fixpoint algorithm (Pötzelberger and Strasser, definition 4.26) finally reads as follows.

Definition 5. *(Fixpoint algorithm)*
Let $\mathbf{a} \in A^m$ *be a set of derivatives of the convex function* f*. Get a new set* $\mathbf{b} = T(\mathbf{a})$ *by the following steps:*

1. *Choose any MSP-partition* $\mathcal{B} \in \mathcal{S}(\mathbf{a})$ *for which* $I_f(\mathcal{B})$ *is maximal.*
2. *If* $\mathcal{B}$ *contains no set of* P*-measure zero, choose* $\mathcal{B}^\star := \mathcal{B}$*, otherwise as long as this condition is not true continue splitting the set* B_k *with the highest probability in two sets with positive* P*-measure while increasing each time the* f*-information.*
   Compute the means $m(B_k)$ *of the sets in the partition* $\mathcal{B}^\star$*.*
3. *Choose any* b_k *from the subdifferential* $D(f, m(B_k))$ *and define* $T(\mathbf{a}) := \mathbf{b} = (b_1, b_2, \ldots, b_m)$*.*

The subdifferential $D(f, x)$ is the set of all derivatives respective slopes of the maximal support functions $l(x)$ with $l \leq f$ on $\mathbb{R}^d$ and $l(x) = f(x)$. If f is differentiable then $D(f, x)$ contains only the derivative of f in x.

Note that for $f(x) = |x|^2/2$ at any point f(x) and its derivative coincide such that $b_k = m(B_k)$ holds. The third iteration step drops out and doesn't occur in the k-means algorithm.

Since we apply the algorithm to data drawn from Lebesque-continuous distributions and to differentiable convex functions only, the MSP-partition in step 1. and the differential in step 3. are uniquely determined with probability 1. The splitting in step 2 is done by choosing a random data point out of the partition to be splitted. The algorithm has been implemented in the Doctoral thesis of Steiner (1999), who experimentally treated different questions concerning this data compression method, among others the determination of a suitable number of representing points, the choice of the convex function and improvements of the algorithm. In chapter 4 we state the exact implementation of the algorithm used for our simulations with discrete data sets.

Pötzelberger and Strasser (1999) showed that the fixpoint algorithm converges to a local optimum of the optimization problem (3), which is for discrete data sets a simple consequence of the increase of the information measure $I_f(\mathcal{B}) := \sum_{j=1}^{m} P(B_j)\, f(m(B_j))$ (1) in any iteration step. For further technical details and theoretical results we again refer to this article.

Figure 2 shows one dimensional plots of the four convex functions we use in our simulation studies.

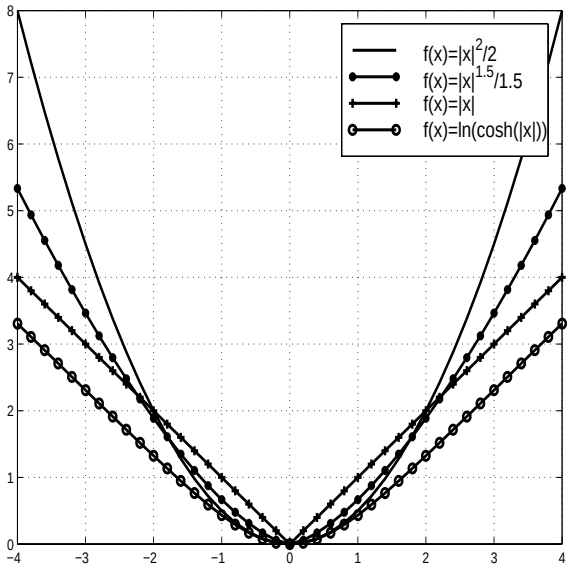

Figure 2: Convex functions used for data compression

To illustrate the differences of partitions obtained with these functions we plot representations of typical partitions in two dimensions, figures 3 and 4. These partitions are the result of the application of the fixpoint algorithm to 400 standard normal distributed data.

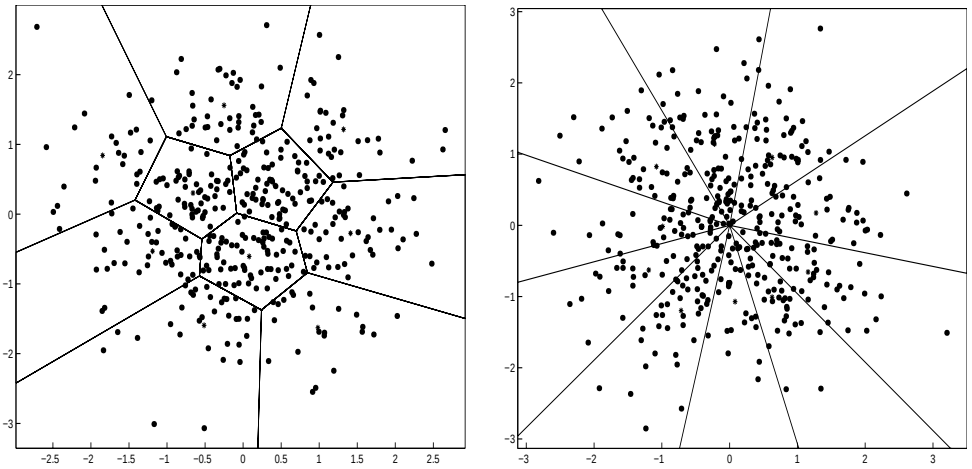

Figure 3: MSP-partitions for $f(x) = |x|^2/2$ (left) and $f(x) = |x|$ (right)

Whereas $f(x) = |x|^2/2$ leads to a k-means type segmentation, $f(x) = |x|$ again results in a Kohonen-type directional based partition.

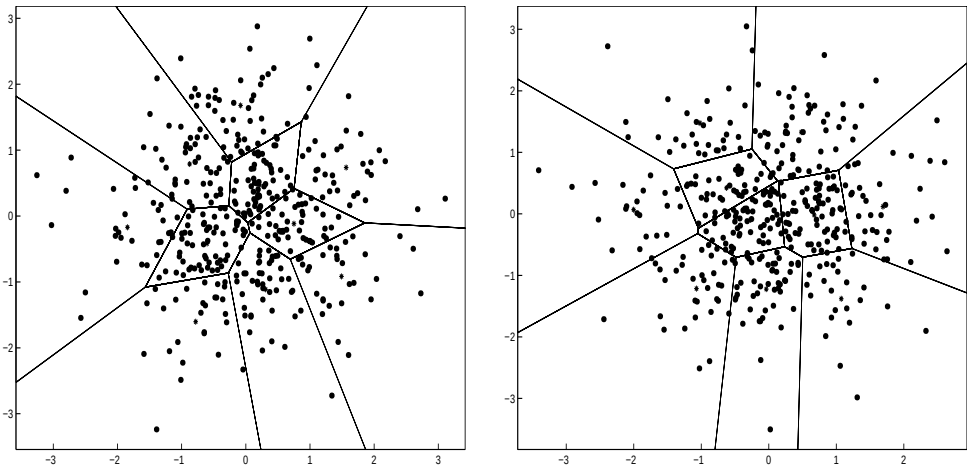

Figure 4: MSP-partitions for $f(x) = |x|^{1.5}/1.5$ (left) and $f(x) = \ln\cosh(|x|/2)$ (right)

The function $f(x) = |x|^{1.5}/1.5$ is the compromise noted before in one dimension and the application of $f(x) = \ln\cosh(|x|/2)$ is another valuable combination, behaving like $|x|^2/2$ in the interior and like $|x|$ in the outer part which can be recognized also in figure 2 from the shape of this function. These observations will guide our investigations in chapter 4.

3 Permutation tests for the k-sample problem

To compare the properties and the usefulness of the data compression methods illustrated in the preceeding chapter we use the most popular multivariate testing problem, the k-sample problem, for which test theory is well worked out. We first compress the data sets and then evaluate the tests for the clustered data. A general formulation of the test problem is the following.

Definition 6. *(k-sample problem):*
For fixed numbers $k \in \mathbb{N}$ and $n_i \in \mathbb{N}$, $1 \le i \le k$, let $X_{i1}, \ldots, X_{in_i}$, $1 \le i \le k$ be independent identically distributed random variables with distribution P_i. Define the null hypothesis H_0 and the alternative H_A by

$$H_0 := \{P_i = P_j \,|\, 1 \le i, j \le k\} \tag{8}$$

$$H_A := \{P_i \ne P_j \,|\, \exists\, i \ne j\} \tag{9}$$

To define the test statistics we need some further denotations.

Definition 7. *For fixed numbers $k, n_i \in \mathbb{N}$, $1 \leq i \leq k$ let $(x_{il})_{1\leq i\leq k, 1\leq l\leq n_i}$ be a multivariate data set with $n := \sum_{1=1}^{k} n_i$ observations in total. Let $\mathcal{B} = (B_1, \ldots, B_m)$ be a partition that indicates a segment membership resulting from a clustering algorithm. The number of points in B_j and their midpoint are denoted by*

$$h_j := \#\{x_{il} \in B_j : 1 \leq i \leq k, 1 \leq l \leq n_i\}, \tag{10}$$

$$m_j := \frac{1}{h_j} \sum_{x_{il}\in B_j} x_{il}, \tag{11}$$

and the number of points in B_j, restricted to sample i by

$$h_j^{(i)} := \#\{x_{il} \in B_j : 1 \leq l \leq n_i\}. \tag{12}$$

A common test statistic for the k-sample problem is the well known variance quotient statistic which is asymptotically optimal for normal distributions (cf. Witting (1985), p. 220). Precisely this is the quotient of the variance of the midpoints of the samples devided by an overall variance estimator of the pooled sample:

$$T(x_{11}, \ldots, x_{kn_k}) = \frac{\frac{1}{k-1} \sum_{i=1}^{k} |\overline{x}_{i\cdot} - \overline{x}_{\cdot\cdot}|^2}{\frac{1}{n-k} \sum_{i=1}^{k} \sum_{l=1}^{n_i} |x_{il} - \overline{x}_{i\cdot}|^2}, \tag{13}$$

where

$$\overline{x}_{i\cdot} = \frac{1}{n_i} \sum_{l=1}^{n_i} x_{il}, \qquad \overline{x}_{\cdot\cdot} = \frac{1}{n} \sum_{i=1}^{k} \sum_{l=1}^{n_i} x_{il} \tag{14}$$

are the center of the i-th and the pooled sample.

We apply this statistic for the clustered data in the following way. As a result of the data compression method every point $x_{il} \in B_j$ of the original data set is viewed as a representative of the corresponding midpoint m_j of all points in B_j and replaced by it. The modified test statistic then reads as

$$T_1^\star(x_{11}, \ldots, x_{kn_k}) = \frac{\frac{1}{k-1} \sum_{i=1}^{k} \left|m^{(i)} - \overline{m}\right|^2}{\frac{1}{n-k} \sum_{i=1}^{k} \sum_{j=1}^{m} h_j^{(i)} |m_j - m^{(i)}|^2}. \tag{15}$$

with the center $m^{(i)}$ of the modified i-th sample and the overall center $\overline{m}$, defined by

$$m^{(i)} = \frac{1}{n_i} \sum_{j=1}^{m} h_j^{(i)} m_j, \qquad \overline{m} = \frac{1}{k} \sum_{i=1}^{k} m^{(i)}. \tag{16}$$

To calculate $m^{(i)}$ we have to take the mean of the segmentation midpoints m_j, weighted by their frequency $h_j^{(i)}$ in sample i.

Usually the data set is standardized before applying the clustering algorithms from chapter 2 which implies $\overline{m} = 0$ since $\overline{m} = \overline{x}_{..}$. Another simplification results from omitting the variance estimator in the denominator in (17) to save runtime. This can be done since we know and control the variance within the samples. Some simulations will also analyze the numerical differences. Therefore we define

$$T_1(x_{11}, \ldots, x_{kn_k}) \quad = \quad \frac{1}{k-1} \sum_{i=1}^{k} \left| m^{(i)} - \overline{m} \right|^2 . \tag{17}$$

The second, competing test statistic is of typical χ^2-type, precisely

$$T_2(x_{11}, \ldots, x_{kn_k}) \quad = \quad \sum_{i=1}^{k} \sum_{j=1}^{m} \frac{|h_j^{(i)} - n_i\, h_j|^2}{n_i\, h_j} \tag{18}$$

and simply measures the homogenity of the frequencies of points with respect to sample and cluster membership. The values of the midpoints are not considered.

Because of the nonparametric nature of the clustering algorithm we don't know the exact and not even an asymptotic distribution of the test statistic even under the null hypothesis. To assure that the tests nevertheless hold the desired significance level we have to apply multivariate permutation tests. This means that first the test statictic is calculated for all random permutations of the original or equivalently of the modified, clustered data. The test then rejects the null hypothesis if the value of the test statistic is greater than a portion of $1-\alpha$ of the values of the test statistics with the permuted data, where α indicates the significance level:

Definition 8. *(permutation test):*
Let $\alpha \in [0,1] \subset \mathbb{R}$ be fixed and the test statistics T_1 and T_2 be defined by the formulas (17) and (18). Then for $i = 1, 2$ we define the permutation tests φ_i by

$$\varphi_i \quad = \quad \begin{cases} 1, & > \\ & T_i \qquad c_i(\alpha), \\ 0, & \leq \end{cases} \tag{19}$$

where $c_i(\alpha)$ is the $1-\alpha$-quantile of the distribution of $T_i(\sigma(X))$ for fixed $X = (x_{11}, \ldots, x_{kn_k})$ and σ uniformly distributed on the set of all permutations of the numbers $1, \ldots, n$ with $n = \sum_{i=1}^{k} n_i$.

A supporting justification of the use of permutation tests in this context and more theoretical background can be found in a paper of Strasser and Weber

(1999) who obtained asymptotical optimality results of such tests with the help of LAN-theory.

Let us mention that the present approach can also be regarded as a multivariate extension of the theory of rank tests in the one dimensional case, where a data point is replaced in the test statistic by the expectation of its corresponding rank in the sample, see e.g. Hájek (1969), chapter III, IV, who demonstrated the dominating influence of the existance of outliers for the optimal choice of the scores. The effect of tail dependence of optimal test procedures is well known in the field of rank tests and of great importance in the field of robust statistics.

4 Simulations

We first describe the clustering algorithm used for data compression in our simulation studies and then list and explain the fixed and variable parameters.

Let X be the multivariate data set with $n \in I\!N$ observations. The implementation of the fixpoint algorithm is the following.

Definition 9. *(Implementation of fixpoint algorithm)*
First select a random sample of m initial prototypes from X. Then repeat the following steps until the algorithm terminates.

- *Evaluate the maximal support functions.*
- *Determine the corresponding MSP-partition $\mathcal{B} = (B_1, \ldots, B_m)$.*
- *If $\mathcal{B}$ contains partitions with no points, continue splitting the set B_k with the highest number of points h_k in two sets by replacing the prototype with no representing points by a random point of B_k.*
- *Calculate the midpoints c_i of the new partition which represent the new prototypes.*
- *Calculate the new value of the f-Information*

$$I_f \;=\; \frac{1}{N} \sum_{i=1}^{m} h_i \, f(c_i),$$

and stop if there is no improvement.

The algorithm stops after a finite number of iteration steps, because for a discrete data set there exists only a finite number of possible segmentations and the f-Information is strictly increasing in each iteration step by construction.

Fixed paramaters occuring in our simulations are:

parameter	value	description
level	0.05	level of the test
alternatives	10	number of equidistant alternatives for which the power is calculated
permiter	1000	number of permutations carried out to calculate the critical value of the permutation test
geniter	1000	number of Monte Carlo replications to calculate the power function for a fixed alternative

Variable parameters are:

parameter	values	description
k	2,3,10	number of samples
distribution	N	standard normal
	E	standard exponential (radius exp., angle unif.)
	M	mixture of normal (90%) and exponential (10%)
	U	uniform on $\{x : \|x\| \leq 1\}$
	C	cauchy (radius cauchy, angle uniform)
freq	100,200	data points per sample (200 for k=2, 100 otherwise)
dim	2,5	dimension
m	6,10,20	number of prototypes resp. clusters
p	0,1,1.5,2	parameter for convex function f (see below)

For the multivariate standard exponential and the cauchy distribution a data point is drawn by first choosing a direction (angle) uniformly distributed on the sphere and then the distance from the origin which is exponentially resp. cauchy distributed either with parameter 1.

The following four convex functions are considered (see figure 2).

$$\begin{aligned} f_p(x) &:= |x|^p/p, \qquad\qquad p = 1,\ 1.5,\ 2, \\ f_0(x) &:= \ln\left(\cosh\left(\frac{|x|}{2}\right)\right). \end{aligned}$$

The k samples are defined as shift alternatives. In the 'linear' case the sample midpoints of the alternatives lie with equal distances on a line in the direction of the first coordinate, for 'circle' alternatives the sample midpoints lie equidistant on a circle in the first two coordinates.

The scale in the calculations of the power functions is given by the difference of two adjacent samples in the linear case and by the radius in the circlular case.

As test statistics T_1 and T_2, formulas (17) and (18) are applied, the default is the statistic T_1.

Figure 5 shows typical resulting MSP-partitions of the fixpoint algorithm for two two dimensional normal resp. exponential distributed samples of size 200 for the parameters $p = 2$ and $p = 1$. We see that in the exponential case and for $p = 2$ the outliers (left bottom) strongly influence the partition which will be reflected in the power results.

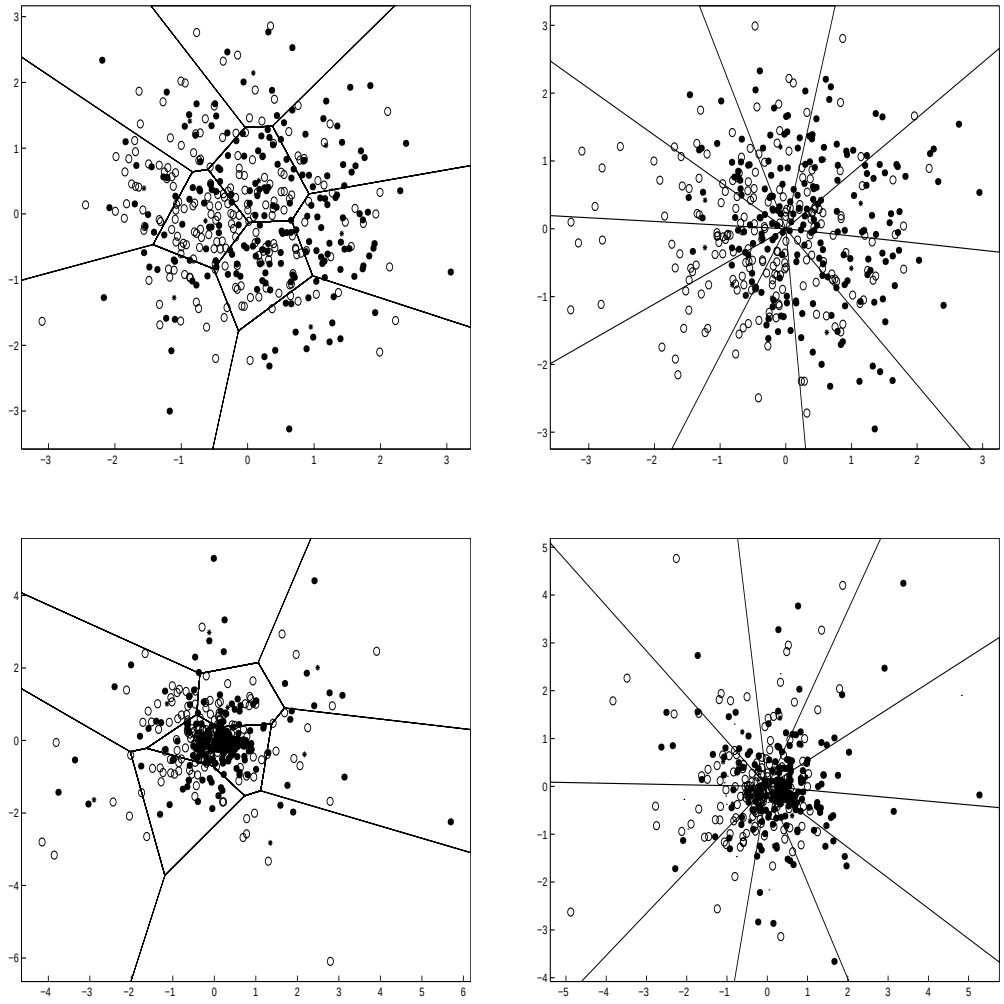

Figure 5: Resulting MSP-partitions for normal (top) and exponential (bottom) distributions for $p = 2$ (left) and $p = 1$ (right)

5 Results

We now present exemplary pictures of simulated power functions and the tables with more detailed results. In the tables the samples are quoted in the suggestive form *k*distribution(freq,dim)*, for example $2 * E(100, 5)$ stands for two exponential samples with 100 observations per sample in five dimensions.

Pictures 6 to 8 show results for normal distributions. The k-means type partition with $p = 2$ and the balancing versions with $p = 1.5$ and $p = 0$ ($\ln\cosh$) produce similar outcomes, but a Kohonen-type partition with $p = 1$ leads to a loss of power. This loss grows up to 0.1 in the case of $k = 10$ samples (picture 8). A closer look at tables 1 and 2 indicates that this effect can be observed for nearly all parameter constellations, only in the tested five dimensional case there are no significant differences. Picture 6 and 7 show that using the χ^2-test statistic (18) leads to a lower power for all compression methods but retains the order of the quality of the diverse methods.

If the original data are uniformly distributed there are no outliers and the described effect becomes even more striking (picture 9). Using 20 prototypes instead of 10 leads to only slightly higher power values (table 3).

For exponentially distributed data pictures 10 and 11 demonstrate that the situation is upside down, but in a more drastic manner. The Kohonen type partition is in all simulations far superior to the others in detecting differences of the k samples. The amount of power gain is always enormous and reaches values up to 0.7 (table 4), these differences can also be found for five dimensional data. Tests with cauchy distributions (picture 12, table 5) show that for distributions with even more extreme outliers, when the k-sample problem is naturally complicate to handle, the Kohonen algorithm again provides the best results, clearly separated from the others. There is practically no difference using 10 or 20 prototypes (table 5).

The last example deals with mixture distributions where 90% of the data are drawn from a standard normal distribution and 10% from an exponential distribution. The low number of extreme data already suffices to favour the Kohonen algorithm for solving the k-sample problem (picture 13, table 6).

In tabular 7 for some normal and exponential distributions a comparison of the application of the test statistics T_1 and $T_1^\star$ (formulas (17) and (15)) is given. Using $T_1^\star$ means in principle also permuting the data in the variance estimator in the denominater of the original test statistic. For normal data there are practically no differences to observe and for exponential data there is a slight gain of power using $T_1^\star$, but the order of the quality of the algorithms remains unchanged.

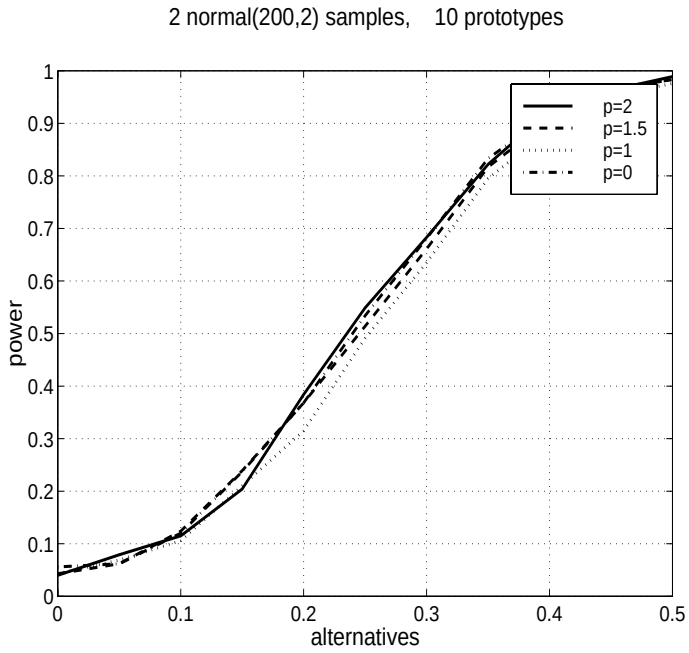

Figure 6: 2 normal distributions of dimension 2, compressed to 10 prototypes, test statistic T_1

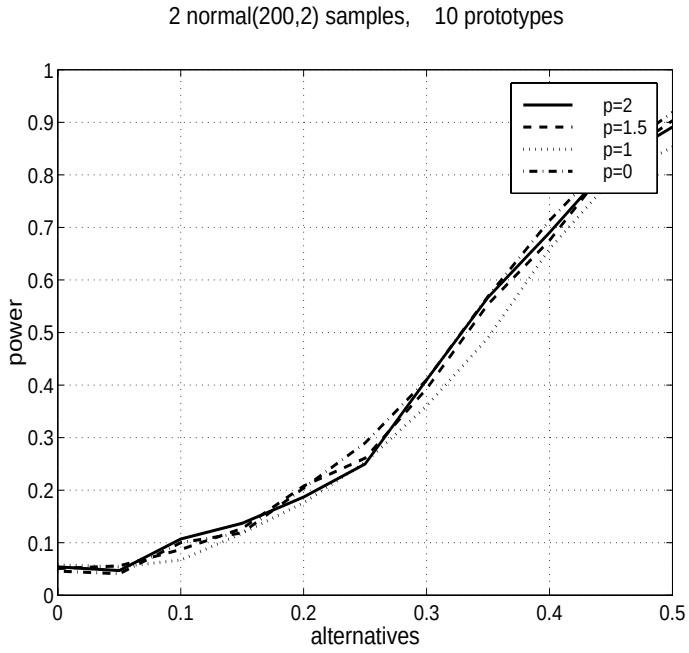

Figure 7: 2 normal distributions of dimension 2, compressed to 10 prototypes, χ^2-test statistic T_2

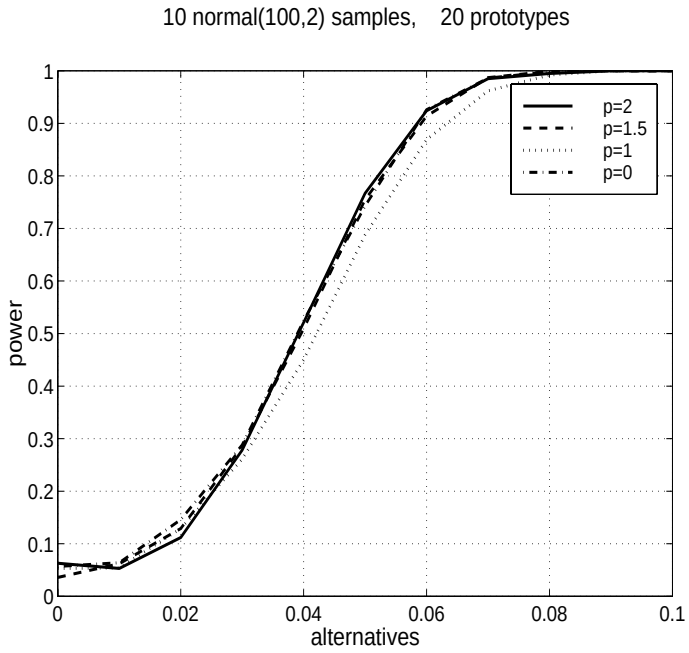

Figure 8: 10 normal distributions of dimension 2, compressed to 20 prototypes

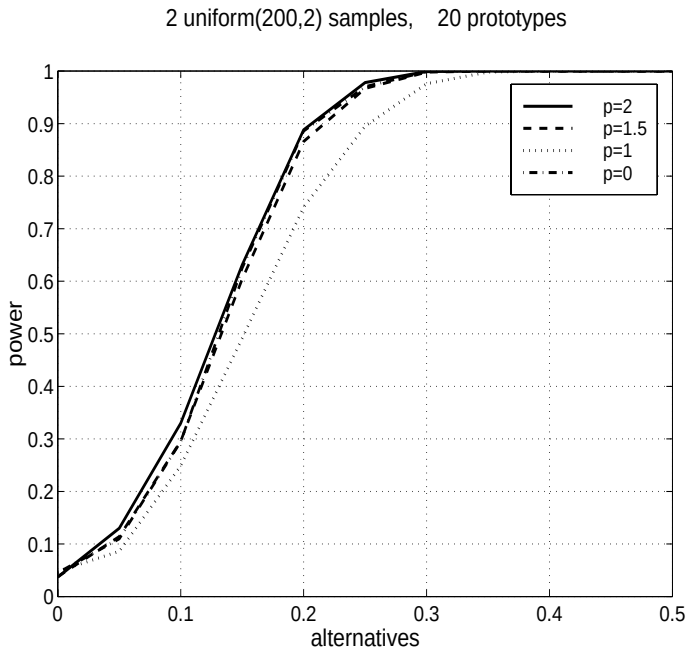

Figure 9: 2 uniform distributions of dimension 2, compressed to 20 prototypes

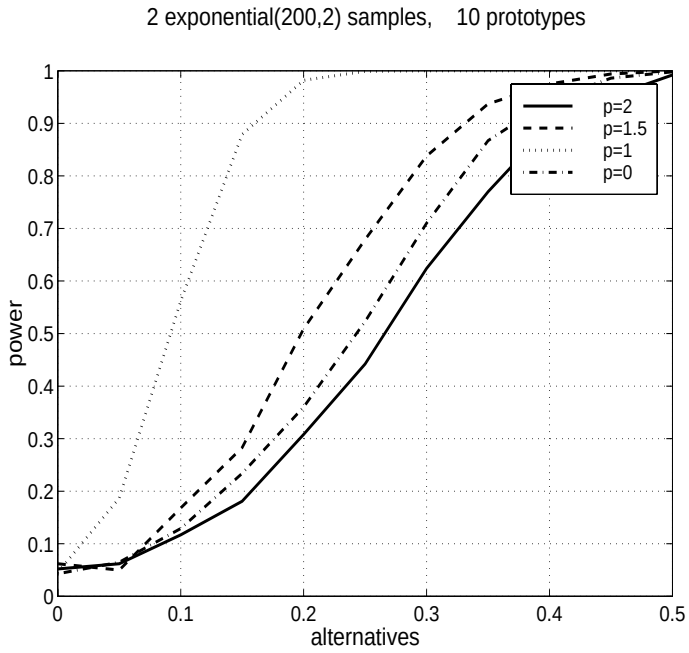

Figure 10: 2 exponential distributions of dimension 2, compressed to 10 prototypes

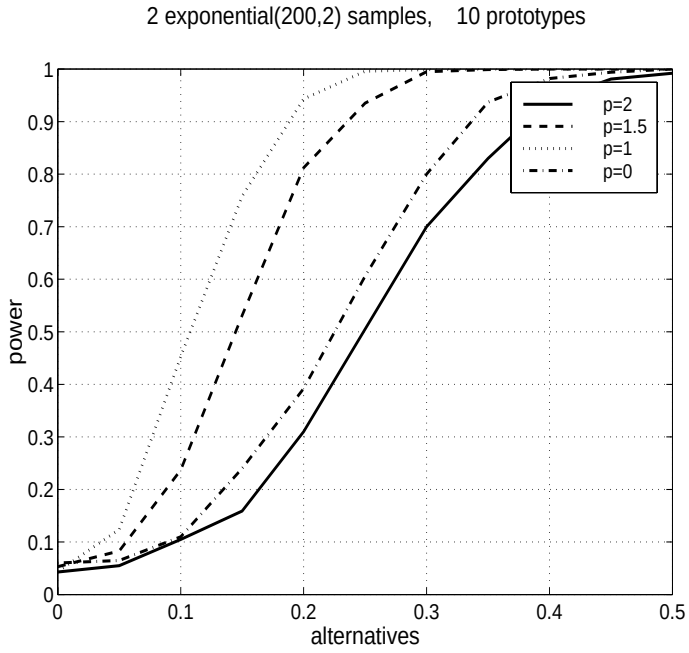

Figure 11: 2 exponential distributions of dimension 2, compressed to 10 prototypes, χ^2-test statistic T_2

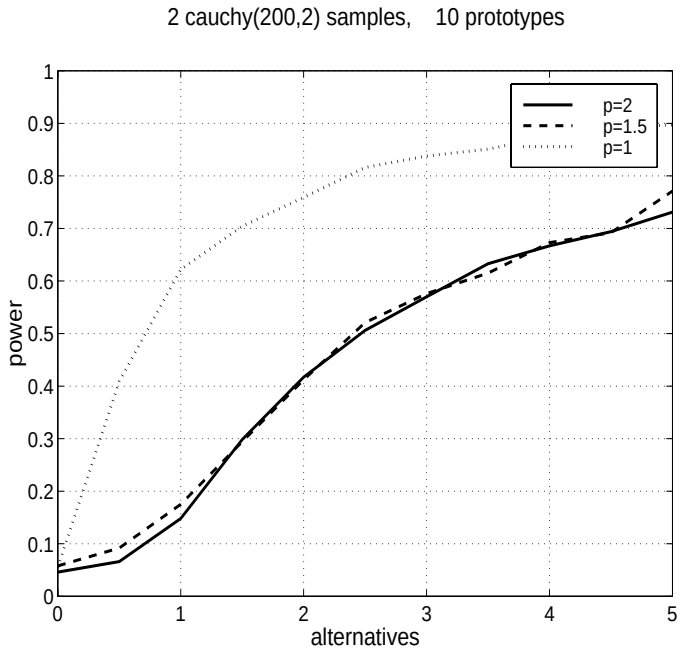

Figure 12: 2 cauchy distributions of dimension 2, compressed to 10 prototypes

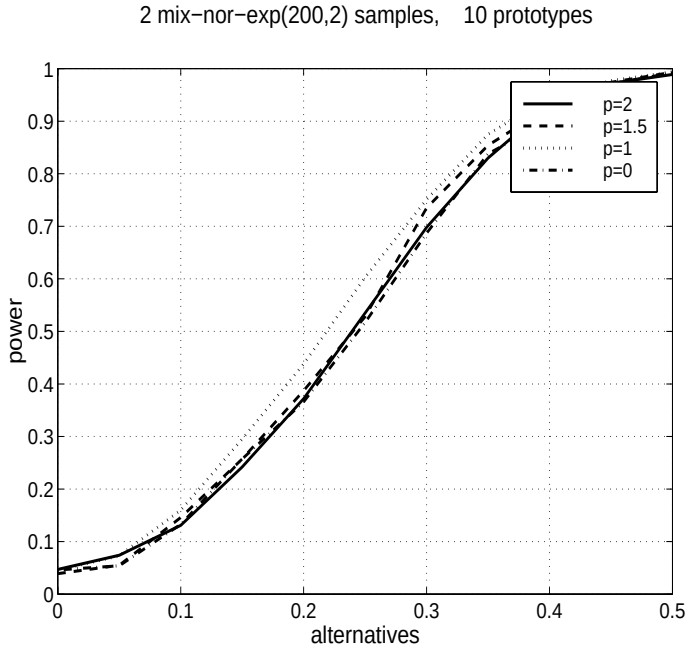

Figure 13: 2 mixture distributions (90 percent normal, 10 percent exponential) of dimension 2, compressed to 10 prototypes

SAMPLES	m	p	Alternatives					
			0 (H_0)	0.05	0.10	0.15	0.20	0.25
2*N(200,2)	6	2	0.051	0.061	0.118	0.210	0.319	0.483
		1.5	0.043	0.068	0.121	0.213	0.337	0.480
		1	0.054	0.062	0.118	0.207	0.313	0.466
		0	0.049	0.065	0.127	0.207	0.353	0.482
2*N(200,2)	10	2	0.040	0.079	0.115	0.204	0.384	0.549
		1.5	0.043	0.062	0.124	0.239	0.368	0.514
		1	0.043	0.069	0.106	0.210	0.315	0.419
		0	0.056	0.062	0.120	0.237	0.368	0.534
2*N(200,2) (χ^2-test φ_2)	10	2	0.054	0.047	0.107	0.137	0.187	0.250
		1.5	0.051	0.056	0.087	0.127	0.208	0.261
		1	0.058	0.054	0.067	0.118	0.176	0.253
		0	0.046	0.041	0.100	0.119	0.204	0.290
3*N(100,2)	10	2	0.073	0.060	0.152	0.296	0.535	0.734
		1.5	0.045	0.076	0.156	0.278	0.502	0.731
		1	0.046	0.071	0.140	0.271	0.448	0.673
		0	0.056	0.078	0.162	0.291	0.519	0.726
3*N(100,2) (circular)	10	2	0.058	0.068	0.079	0.132	0.224	0.325
		1.5	0.051	0.054	0.095	0.123	0.223	0.287
		1	0.053	0.051	0.077	0.112	0.210	0.289
		0	0.058	0.057	0.084	0.151	0.235	0.326
			Alternatives					
			0 (H_0)	0.01	0.02	0.03	0.04	0.05
10*N(100,2)	20	2	0.063	0.053	0.112	0.278	0.521	0.767
		1.5	0.036	0.062	0.129	0.285	0.509	0.755
		1	0.053	0.054	0.129	0.261	0.450	0.688
		0	0.057	0.064	0.146	0.288	0.524	0.743
			Alternatives					
			0 (H_0)	0.1	0.2	0.3	0.4	0.5
2*N(200,5)	10	2	0.043	0.069	0.156	0.341	0.601	0.782
		1.5	0.044	0.082	0.188	0.332	0.568	0.797
		1	0.047	0.081	0.164	0.336	0.550	0.802
		0	0.052	0.085	0.155	0.305	0.556	0.785

Table 1: Simulated power functions for normal distributions, part I: alternatives close to H_0

SAMPLES	m	p	Alternatives					
			0.25	0.30	0.35	0.40	0.45	0.50
2*N(200,2)	6	2	0.483	0.623	0.768	0.873	0.941	0.978
		1.5	0.480	0.603	0.767	0.875	0.944	0.966
		1	0.466	0.605	0.765	0.882	0.927	0.965
		0	0.482	0.632	0.778	0.855	0.938	0.979
2*N(200,2)	10	2	0.549	0.683	0.822	0.924	0.963	0.989
		1.5	0.514	0.661	0.817	0.907	0.967	0.983
		1	0.419	0.635	0.795	0.888	0.952	0.976
		0	0.534	0.681	0.833	0.917	0.965	0.983
2*N(200,2) (χ^2-test φ_2)	10	2	0.250	0.410	0.567	0.690	0.821	0.891
		1.5	0.261	0.393	0.554	0.675	0.823	0.903
		1	0.253	0.360	0.490	0.659	0.790	0.853
		0	0.290	0.410	0.569	0.713	0.834	0.919
3*N(100,2)	10	2	0.734	0.891	0.954	0.993	0.999	1.000
		1.5	0.731	0.886	0.962	0.992	0.999	1.000
		1	0.673	0.837	0.958	0.986	0.991	1.000
		0	0.726	0.882	0.957	0.990	0.999	1.000
3*N(100,2) (circular)	10	2	0.325	0.436	0.607	0.724	0.825	0.907
		1.5	0.287	0.452	0.571	0.720	0.816	0.909
		1	0.289	0.397	0.536	0.660	0.787	0.856
		0	0.326	0.458	0.605	0.705	0.820	0.893
			Alternatives					
			0.05	0.06	0.07	0.08	0.09	0.10
10*N(100,2)	20	2	0.767	0.924	0.985	0.995	1.000	1.000
		1.5	0.755	0.914	0.987	0.999	1.000	1.000
		1	0.688	0.869	0.962	0.992	1.000	1.000
		0	0.743	0.926	0.986	0.998	1.000	1.000
			Alternatives					
			0.5	0.6	0.7	0.8	0.9	1.0
2*N(200,5)	10	2	0.782	0.941	0.985	0.995	0.998	1.000
		1.5	0.797	0.928	0.985	0.996	0.999	1.000
		1	0.802	0.941	0.980	0.995	1.000	1.000
		0	0.785	0.944	0.975	0.996	0.999	1.000

Table 2: Simulated power functions for normal distributions, part II: alternatives distant from H_0

SAMPLES	m	p	Alternatives					
			0 (H_0)	0.05	0.10	0.15	0.20	0.25
2*U(200,2)	10	2	0.046	0.111	0.263	0.559	0.816	0.951
		1.5	0.047	0.102	0.267	0.528	0.758	0.922
		1	0.055	0.091	0.220	0.425	0.696	0.873
		0	0.055	0.093	0.257	0.541	0.783	0.932
2*U(200,2)	20	2	0.037	0.130	0.330	0.631	0.888	0.978
		1.5	0.038	0.114	0.296	0.604	0.866	0.967
		1	0.048	0.086	0.249	0.491	0.741	0.895
		0	0.045	0.110	0.295	0.625	0.886	0.970
			Alternatives					
			0.25	0.30	0.35	0.40	0.45	0.50
2*U(200,2)	10	2	0.951	0.992	0.999	1.000	1.000	1.000
		1.5	0.922	0.985	0.999	1.000	1.000	1.000
		1	0.873	0.976	0.995	0.998	1.000	1.000
		0	0.932	0.988	1.000	1.000	1.000	1.000
2*U(200,2)	20	2	0.978	0.999	1.000	1.000	1.000	1.000
		1.5	0.967	0.998	1.000	1.000	1.000	1.000
		1	0.895	0.976	0.998	1.000	1.000	1.000
		0	0.970	0.999	1.000	1.000	1.000	1.000

Table 3: Simulated power functions for uniform distributions

SAMPLES	m	p	Alternatives					
			0 (H_0)	0.05	0.10	0.15	0.20	0.25
2*E(200,2)	10	2	0.052	0.062	0.117	0.181	0.308	0.442
		1.5	0.062	0.050	0.168	0.283	0.509	0.679
		1	0.045	0.187	0.562	0.879	0.982	0.999
		0	0.043	0.065	0.129	0.234	0.360	0.523
2*E(100,2) (χ^2-test φ_2)	10	2	0.043	0.055	0.105	0.159	0.310	0.505
		1.5	0.053	0.083	0.237	0.531	0.812	0.935
		1	0.044	0.124	0.453	0.758	0.943	0.996
		0	0.060	0.065	0.110	0.240	0.392	0.606
3*E(100,2) (circular)	10	2	0.056	0.063	0.063	0.121	0.173	0.265
		1.5	0.042	0.058	0.091	0.183	0.267	0.434
		1	0.056	0.142	0.364	0.680	0.890	0.966
		0	0.047	0.069	0.076	0.117	0.217	0.303
2*E(200,5)	10	2	0.047	0.052	0.081	0.127	0.284	0.441
		1.5	0.056	0.075	0.132	0.302	0.608	0.846
		1	0.050	0.208	0.751	0.968	0.996	1.000
		0	0.052	0.056	0.094	0.178	0.339	0.573
			Alternatives					
			0.25	0.30	0.35	0.40	0.45	0.50
2*E(100,2)	10	2	0.442	0.624	0.770	0.895	0.951	0.992
		1.5	0.679	0.838	0.937	0.975	0.994	1.000
		1	0.999	1.000	1.000	1.000	1.000	1.000
		0	0.523	0.710	0.867	0.943	0.986	0.998
2*E(100,2) (χ^2-test φ_2)	10	2	0.505	0.700	0.830	0.939	0.981	0.992
		1.5	0.935	0.995	0.999	1.000	1.000	1.000
		1	0.996	0.999	1.000	1.000	1.000	1.000
		0	0.606	0.800	0.937	0.982	0.994	1.000
3*E(100,2) (circular)	10	2	0.265	0.387	0.526	0.671	0.823	0.885
		1.5	0.434	0.594	0.745	0.868	0.935	0.976
		1	0.966	0.996	1.000	1.000	1.000	1.000
		0	0.303	0.457	0.616	0.786	0.873	0.944
2*E(200,5)	10	2	0.441	0.636	0.782	0.883	0.945	0.971
		1.5	0.846	0.960	0.993	0.997	0.999	1.000
		1	1.000	1.000	1.000	1.000	1.000	1.000
		0	0.573	0.756	0.881	0.947	0.979	0.989

Table 4: Simulated power functions for exponential distributions

SAMPLES	m	p	Alternatives					
			0 (H_0)	0.5	1.0	1.5	2.0	2.5
2*C(200,2)	10	2	0.046	0.066	0.148	0.298	0.417	0.506
		1.5	0.058	0.092	0.175	0.294	0.412	0.521
		1	0.055	0.409	0.622	0.704	0.759	0.816
2*C(200,2)	20	2	0.045	0.081	0.180	0.303	0.393	0.499
		1.5	0.053	0.073	0.187	0.298	0.421	0.498
		1	0.042	0.386	0.631	0.729	0.776	0.800
			Alternatives					
			2.5	3.0	3.5	4.0	4.5	5.0
2*C(200,2)	10	2	0.506	0.570	0.633	0.667	0.694	0.731
		1.5	0.521	0.576	0.615	0.673	0.692	0.771
		1	0.816	0.838	0.851	0.875	0.892	0.897
2*C(200,2)	20	2	0.499	0.596	0.652	0.666	0.727	0.730
		1.5	0.498	0.567	0.627	0.687	0.728	0.738
		1	0.800	0.833	0.865	0.870	0.883	0.885

Table 5: Simulated power functions for cauchy distributions

SAMPLES	m	p	Alternatives					
			0 (H_0)	0.05	0.10	0.15	0.2	0.25
2*M(200,2)	10	2	0.047	0.074	0.131	0.242	0.372	0.535
		1.5	0.039	0.055	0.146	0.257	0.387	0.528
		1	0.045	0.073	0.160	0.295	0.438	0.603
		0	0.046	0.054	0.133	0.257	0.366	0.517
			Alternatives					
			0.25	0.3	0.35	0.40	0.45	0.50
2*M(200,2)	10	2	0.535	0.698	0.830	0.930	0.972	0.989
		1.5	0.528	0.735	0.854	0.928	0.974	0.993
		1	0.603	0.751	0.875	0.947	0.977	0.994
		0	0.517	0.687	0.838	0.902	0.971	0.992

Table 6: Simulated power functions for mixture distributions (90% normal and 10% exponential)

SAMPLES	m	p	Alternatives					
			0 (H_0)	0.05	0.10	0.15	0.20	0.25
2*E(200,2)	10	2	0.049	0.058	0.075	0.139	0.230	0.341
		2⋆	0.049	0.044	0.109	0.186	0.292	0.448
		1	0.046	0.181	0.543	0.853	0.977	0.996
		1⋆	0.052	0.184	0.592	0.867	0.978	0.996
2*N(200,2)	10	2	0.040	0.079	0.115	0.204	0.384	0.549
		2⋆	0.058	0.066	0.128	0.228	0.368	0.528
		1	0.043	0.069	0.106	0.210	0.315	0.419
		1⋆	0.042	0.059	0.119	0.207	0.332	0.489
			Alternatives					
			0.25	0.30	0.35	0.40	0.45	0.50
2*E(200,2)	10	2	0.341	0.515	0.635	0.726	0.859	0.925
		2⋆	0.448	0.622	0.794	0.911	0.968	0.989
		1	0.996	1.000	1.000	1.000	1.000	1.000
		1⋆	0.996	1.000	1.000	1.000	1.000	1.000
2*N(200,2)	10	2	0.549	0.683	0.822	0.924	0.963	0.989
		2⋆	0.528	0.668	0.817	0.908	0.966	0.988
		1	0.419	0.635	0.795	0.888	0.952	0.976
		1⋆	0.489	0.674	0.789	0.885	0.951	0.978
			Alternatives					
			0 (H_0)	0.01	0.02	0.03	0.04	0.05
10*N(100,2)	20	2	0.063	0.053	0.112	0.278	0.521	0.767
		2⋆	0.053	0.054	0.121	0.275	0.507	0.790
		1	0.053	0.054	0.129	0.261	0.450	0.688
		1⋆	0.051	0.059	0.111	0.235	0.453	0.660
			Alternatives					
			0.05	0.06	0.07	0.08	0.09	0.10
10*N(100,2)	20	2	0.767	0.924	0.985	0.995	1.000	1.000
		2⋆	0.790	0.916	0.986	0.996	1.000	1.000
		1	0.688	0.869	0.962	0.992	1.000	1.000
		1⋆	0.660	0.860	0.962	0.994	1.000	1.000

Table 7: Simulated power functions, comparison of test statistics T_1 (17) and $T_1^\star$ (15) for normal and exponential distributions

6 Conclusions

In the field of data compression the presence of outliers respectively the tails of the underlying distribution of the data naturally are of great importance. The results of our simulations allow the interpretation that this is in fact a decisive determinant to judge the quality of cluster algorithms. For the application of a k-sample problem to compressed data sets the resulting power functions clearly indicate that the tails of the distributions determine the order of the quality of the methods within the chosen class of MSP-generating algorithms.

For data with light tails and only few outliers like in normal or even uniform distributions the popular k-means and the compromising algorithms produce similar results whereas the strictly direction orientated Kohonen type algorithm leads to a significant loss of power. Only for the simulations with five dimensional normal data sets no differences are observed, probably due to the too low number of observations or prototypes for this higher dimension.

The opposite effect can be found for distributions with heavy tails or many outliers, but in a far more drastic manner. On exponential data the Kohonen algorithm performs best and k-means is the worst leading to an enormous, higly significant loss of power. The other algorithms produce results in between. The quality increases with the degree of how directional orientated the algorithm is. For cauchy distributions we observe the familiar difficulties treating a k-sample problem with such data because of the extreme outliers which lead to a dramatic power loss, again the Kohonen algorithm is the only one to accept.

The simulations with mixture distributions of 90 % normal and 10 % exponential data show that even this amount of outliers already suffices to favour the Kohonen algorithm.

Of course other parameters, like the number of prototypes used for the data compression or the test statistic, also influence the look of the power function. But especially the more detailed simulations for normal and exponential distributions tell us that the order of the quality of the methods taken into account is nearly independent of the constellation of other parameters than the tail behaviour.

We thus can summarizing conclude that only if one is sure to have only very few outliers in a data set the usually unscrupulously applied k-means algorithm should be used whereas in any other case a Kohonen type version is to be prefered. This throws afresh a light on this algorithm as a useful tool in robust statistics.

References

Bock, H. H. (1974), *Automatische Klassifikation*, Vandenhoeck und Ruprecht.

Bock, H. H. (1992), A clustering technique for maximizing ϕ-divergence, noncentrality and discriminating power. In M. Schader, editor, *Analyzing and Modeling Data and Knowledge*, p.19–36, Springer.

Flury, B. (1993), Estimation of principal points, *Appl. Statist.*, 42:139-151.

Hájek, J. (1969), *A Course in Nonparametric Statistics*, Holden-Day.

Kohonen, K. (1984), *Self organization and associative memory*, Springer.

Pötzelberger, K. and Strasser, H. (1999), 'Clustering and Quantization by MSP-Partitions', submitted.

Steiner, G. (1999), *Statistical Data Compression by Optimal Segmentation — Theory, Algorithm and Experimental Results*, PhD Thesis, Vienna University of Economics and Business Administration.

Strasser, H. and Weber, C. (1999), 'On the asymptotic theory of permutation statistics', submitted.

Witting, H. (1985), *Mathematische Statistik I*, Teubner.