Journal of Classification 20:77-92 (2003)

Concavity in Data Analysis

Michael P. Windham

University of South Alabama

Abstract: Concave functions play a fundamental role in the structure of and minimization of badness-of-fit functions used in data analysis when extreme values of either parameters or data need to be penalized. This paper summarizes my findings about this role. I also describe three examples where concave functions are useful: building measures of badness-of-fit, building robust M-estimators, and building clustering objective functions. Finally, using the common thread of concavity, all three will be combined to build a comprehensive, flexible procedure for robust cluster analysis.

Keywords: Optimization, Objective functions, Least squares, Partitioning, Badness-of-fit, k-Means, Mixture analysis, Fuzzy clustering, M-Estimators, Robust cluster analysis.

Author's address: Michael P. Windham, Department of Mathematics and Statistics, University of South Alabama, Mobile, AL 36688-0002, USA

1. Introduction

Twenty years ago I gave a lecture on cluster analysis wherein I stated that certain clustering algorithms involved minimizing an objective function built with a concave function. A member of the audience insisted that one does not minimize concave functions. Admitting that this person was correct, I spent the remainder of the year trying to explain to him and to myself why concavity was there and why it was the reason the algorithm worked. Since then I have continued to work in cluster analysis and also in robust statistical estimation and optimization. I have found many times that *concave* functions have played a fundamental role in the *structure* of and *minimization* of badness-of-fit functions used in these areas of data analysis. This paper describes that role.

As examples, I will describe three instances of the use of concave functions: building measures of badness-of-fit, building robust M-estima tors, and building clustering objective functions. In each of these cases extreme values of either parameters or data need to be penalized. I will discuss how and why concavity is involved in the penalizing process. Finally, I will put all three together to build a comprehensive, flexible fit criterion for robust cluster analysis.

Concavity also facilitates the minimization of the fit criteria using iterative majorization (Heiser, 1995) as will be described in Section 7.

This paper organizes and unifies the common thread of concavity that runs through the several examples. The role of concavity has not always been recognized in data analysis, and a fundamental role it is. The latter is the main point of this paper.

2. Concave Functions

Concave functions are involved in data analysis in reducing the influence of extreme values of some measure of fit and optimizing an objective function built from the measure. Before examining specific uses of concave functions, I will review the basic facts about them that will be useful here.

There are many characterizations of concave functions, but the following is the best for our purposes. A function, $f : R^p \to R$ is *concave* on a set S, if for each $\mathbf{r}$ in S there is a $1 \times p$ matrix, $\mathbf{F}(\mathbf{r})$, satisfying for all $\mathbf{r}$ and $\mathbf{r}_0$ in S

$$f(\mathbf{r}) \leq \mathbf{F}(\mathbf{r}_0)(\mathbf{r} - \mathbf{r}_0) + f(\mathbf{r}_0). \tag{1}$$

When f is differentiable, the matrix $\mathbf{F}(\mathbf{r})$ is its derivative and concavity simply says that the graph of f always lies below its tangent planes. For

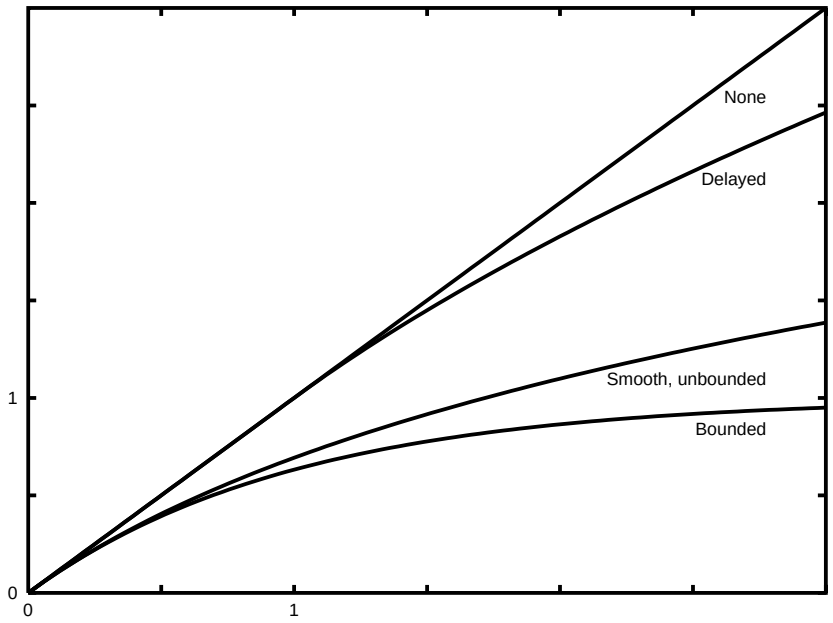


Figure 1: Types of concave functions.

this paper, I will use lower case letters for a concave function and the same upper case letter for the $1 \times p$ matrix valued function required for the linear term in (1).

Another, perhaps more common definition of concavity is that for any $\mathbf{r}_1$ and $\mathbf{r}_2$ in a convex set S and $0 \leq \alpha \leq 1$

$$f(\alpha \mathbf{r}_1 + (1-\alpha)\mathbf{r}_2) \geq \alpha f(\mathbf{r}_1) + (1-\alpha) f(\mathbf{r}_2)$$

That is, the value of f on a convex combination of vectors exceeds the convex combination of the values.

Suppose that r represents a measure of fit between data and parameters describing data for which large values suggest poor fit. The effect of concave functions on reducing the influence of extreme values is shown in Figure 1. In each case, $f(r)$ increases as r does, but $f(r) \leq r$ and the difference increases as r increases, so that the influence of large values of r is reduced.

In building the measures of fit themselves, Section 3, concavity is used to penalize extreme values of the parameters in the measure, so that desirable minima exist. In robust estimation, Section 4, concavity reduces the influence of outliers, and in cluster analysis, Section 5, concavity localizes on a cluster and reduces the influence in describing the cluster of data from other clusters.

3. Parameter Estimation Using Measures of Fit

The basic goal of exploratory data analysis is to summarize data with manageable quantities that describe some informative characteristics of the data. For example, for a set X of n data points in R^d, the mean $\bar{\mathbf{x}} = \sum_{\mathbf{x}} \mathbf{x}/n$ gives a "central" location for the data, and the covariance or scatter matrix, $\hat{\mathbf{S}} = \sum_{\mathbf{x}} (\mathbf{x} - \bar{\mathbf{x}})(\mathbf{x} - \bar{\mathbf{x}})^T/n$ describes the shape of the data set in terms of ellipsoids determined by a "natural" choice of a metric, $(\mathbf{x} - \bar{\mathbf{x}})^T \hat{\mathbf{S}}^{-1} (\mathbf{x} - \bar{\mathbf{x}})$.

It is even more satisfying when a "natural" choice is a "best" choice in some sense. For example, the squared Euclidean norm, $|\mathbf{x} - \mathbf{m}|^2$, is a measure of the fit between a data point $\mathbf{x}$ with a possible center of concentration $\mathbf{m}$, in that the further $\mathbf{m}$ is from $\mathbf{x}$, the more incompatible they are with each other. It is well-known that the mean is the value of $\mathbf{m}$ that minimizes $\sum_{\mathbf{x}} |\mathbf{x} - \mathbf{m}|^2$, that is $\bar{\mathbf{x}}$ is a "best" choice for a center of a data set in that it is on the average least incompatible with the data in the set.

It would be nice to have a similar measure of fit that leads to $\hat{\mathbf{S}}$ being a best choice. A natural candidate would be to minimize $d(\mathbf{m}, \mathbf{S}) = \sum_{\mathbf{x}} (\mathbf{x} - \mathbf{m})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{m})$, particularly since for any positive semidefinite $\mathbf{S}$, we have $d(\bar{\mathbf{x}}, \mathbf{S}) \leq d(\mathbf{m}, \mathbf{S})$ for all $\mathbf{m}$. It suffices then to minimize $d(\bar{\mathbf{x}}, \mathbf{S}) = \sum_{\mathbf{x}} (\mathbf{x} - \bar{\mathbf{x}})^T \mathbf{S}^{-1} (\mathbf{x} - \bar{\mathbf{x}}) = n \operatorname{tr}(\hat{\mathbf{S}}\mathbf{S}^{-1})$, where $\operatorname{tr}(\cdot)$ denotes the trace of a matrix. Unfortunately, the function d has no minimum as a function of $\mathbf{S}$. The problem is that the larger $\mathbf{S}$ is (in terms of say eigenvalues), the smaller d is. People have added "penalty terms" to the measure to discourage large values of scale. Finding such terms leads to the first use of concavity, namely *using concave functions to penalize large values of scale in measuring fit.*

The first step is to describe how to extend a function $g : R \to R$ to a function assigning symmetric matrices to symmetric matrices. The function g is extended to a symmetric matrix $\mathbf{S}$ by letting $\mathbf{g}(\mathbf{S}) = \mathbf{B} \operatorname{diag}(g(\lambda_1), \ldots, g(\lambda_d))\mathbf{B}^T$, where $\mathbf{B}$ is an orthogonal matrix whose columns are a basis of eigenvectors of $\mathbf{S}$ and $\lambda_1, \ldots, \lambda_d$ are the corresponding eigenvalues. Using this notion, we have the following.

If g is concave and increasing, and G satisfies $g(s) \leq G(s_0)(s - s_0) + g(s_0)$ for all s and s_0, then

$$D(\mathbf{m}, \mathbf{S}) = \sum_{\mathbf{x}} \Big[(\mathbf{x} - \mathbf{m})^T \mathbf{G}(\mathbf{S})(\mathbf{x} - \mathbf{m}) + \operatorname{tr}(\mathbf{g}(\mathbf{S}) - \mathbf{G}(\mathbf{S})\mathbf{S}) \Big] \qquad (2)$$

is minimized by $\mathbf{m} = \bar{\mathbf{x}}$ and $\mathbf{S} = \hat{\mathbf{S}}$, the sample mean and covariance matrix. Moreover, if $g(0) \geq 0$ then $(\mathbf{x} - \mathbf{m})^T \mathbf{G}(\mathbf{S})(\mathbf{x} - \mathbf{m}) + \operatorname{tr}(\mathbf{g}(\mathbf{S}) - \mathbf{G}(\mathbf{S})\mathbf{S}) \geq 0$ for all $\mathbf{x}$, $\mathbf{m}$, and positive semidefinite $\mathbf{S}$.

This result was presented in Windham (2000) without proof, so I will give one here.

Clearly, for a fixed, positive semidefinite $\mathbf{S}$, $D(\mathbf{m}, \mathbf{S})$ is minimized for $\mathbf{m} = \bar{\mathbf{x}}$. It suffices, then to show that $D(\bar{\mathbf{x}}, \mathbf{S}) = n \operatorname{tr}(\mathbf{G}(\mathbf{S})(\hat{\mathbf{S}} - \mathbf{S}) + \mathbf{g}(\mathbf{S}))$ is minimized as a function of $\mathbf{S}$ by $\mathbf{S} = \hat{\mathbf{S}}$.

Letting $\hat{\mathbf{\Lambda}}$ be a diagonal matrix of eigenvalues of $\hat{\mathbf{S}}$, it follows by moving orthogonal matrices around in the trace function that, it suffices to show that $\operatorname{tr}(\mathbf{G}(\mathbf{S})(\hat{\mathbf{\Lambda}} - \mathbf{S}) + \mathbf{g}(\mathbf{S})) \geq \sum_i g(\hat{\lambda}_i)$ for all positive semidefinite $\mathbf{S}$. Letting $\mathbf{S} = \mathbf{B}\mathbf{\Lambda}\mathbf{B}^T$ and again moving the orthogonal matrix $\mathbf{B}$, we have from the concavity of g

$$\begin{aligned} \operatorname{tr}(\mathbf{G}(\mathbf{S})(\hat{\mathbf{\Lambda}} - \mathbf{S}) + \mathbf{g}(\mathbf{S})) &= \sum_i G(\lambda_i) \left(\sum_j b_{ij}^2 \hat{\lambda}_j - \lambda_i \right) + g(\lambda_i) \\ &\geq \sum_i g \left(\sum_j b_{ij}^2 \hat{\lambda}_j \right). \end{aligned}$$

Since rows and columns of B are orthonormal vectors, $\sum_i b_{ij}^2 = \sum_j b_{ij}^2 = 1$, so that $\sum_j b_{ij}^2 \hat{\lambda}_j$ is a convex combination of the eigenvalues of $\hat{\mathbf{S}}$. Therefore, from the concavity of the function g, we have $\sum_i g\left(\sum_j b_{ij}^2 \hat{\lambda}_j\right) \geq \sum_i \sum_j b_{ij}^2 g(\hat{\lambda}_j) = \sum_j g(\hat{\lambda}_j)$ and the result follows.

Moreover, if $g(0) \geq 0$, then as above, $(\mathbf{x} - \mathbf{m})^T \mathbf{G}(\mathbf{S})(\mathbf{x} - \mathbf{m}) + \operatorname{tr}(\mathbf{g}(\mathbf{S}) - \mathbf{G}(\mathbf{S})\mathbf{S}) = D(\mathbf{m}, \mathbf{S})$ for a data set consisting of $\mathbf{x}$ alone, so that $D(\mathbf{m}, \mathbf{S})$ is minimized by $\mathbf{m} = \mathbf{x}$ and $\mathbf{S} = \mathbf{0}$. That is, the minimum value of $(\mathbf{x} - \mathbf{m})^T \mathbf{G}(\mathbf{S})(\mathbf{x} - \mathbf{m}) + \operatorname{tr}(\mathbf{g}(\mathbf{S}) - \mathbf{G}(\mathbf{S})\mathbf{S})$ is $D(\mathbf{x}, \mathbf{0}) = g(0) \geq 0$, completing the proof.

It should be noted that since g is increasing, we have $G \geq 0$, so that $\mathbf{G}(\mathbf{S})$ is at least positive semidefinite. Therefore, $(\mathbf{x} - \mathbf{m})^T \mathbf{G}(\mathbf{S})(\mathbf{x} - \mathbf{m}) + \operatorname{tr}(\mathbf{g}(\mathbf{S}) - \mathbf{G}(\mathbf{S})\mathbf{S})$ acts like a metric on the data space.

The goal was to build a fit function that is minimized by the mean and scatter matrix of the data. Any function of the form in (2) is minimized by $\bar{\mathbf{x}}$ as long as $\mathbf{G}(\mathbf{S})$ is positive semidefinite, which requires only that G be non-negative. If so, then one needs to minimize $\operatorname{tr}(\mathbf{G}(\mathbf{S})(\hat{\mathbf{S}} - \mathbf{S}) + \mathbf{g}(\mathbf{S}))$ as a function of $\mathbf{S}$. For simplicity consider the one-dimensional case with g any twice differentiable function. For $d(s) = g'(s)(\hat{s} - s) + g(s)$, we have $d'(s) = g''(s)(\hat{s} - s)$, so that $\hat{s}$ is a critical point. It is the concavity of g that ensures that $\hat{s}$ is, in fact, a minimizer.

For example, with $g(s) = \log(s)$, then $G(s) = 1/s$, and we have the measure $(\mathbf{x} - \mathbf{m})^T \mathbf{S}^{-1} (\mathbf{x} - \mathbf{m}) + \log\det \mathbf{S}$, ignoring constant terms. This

example is popular, but does not satisfy the condition that $g(0) = 0$, which ensures that the measure of fit is non-negative, and facilitates its use in the applications that follow. The function $g(s) = \log(1 + s)$, for example, does satisfy $g(0) = 0$ and yields $(\mathbf{x} - \mathbf{m})^T(\mathbf{I} + \mathbf{S})^{-1}(\mathbf{x} - \mathbf{m}) + \text{tr}(\log(\mathbf{I} + \mathbf{S}) - (\mathbf{I} + \mathbf{S})^{-1}\mathbf{S})$. The second column of Table 1 lists several functions that can be used for g. For functions in the table with two descriptions, except for the Cauchy, the first expression applies for $0 \leq r \leq 1$ and the second for $r > 1$.

We have now a procedure for building badness-of-fit measures that are minimized by the mean and scatter matrix of a data set and that have properties that facilitate their use in data analysis.

4. Robust M-estimation

The notion of choosing descriptors for a data set by minimizing total badness-of-fit between points in the set and some summary descriptor of the data is called M-estimation in the context of robust statistics. Huber (1981) and Hampel, Ronchetti, Rousseeuw, and Stahel (1986) discuss M-estimators in detail and give a variety of procedures for estimating location and shape.

In general, if $\mathbf{t}$ is a descriptor of a data set and $r(\mathbf{x}, \mathbf{t})$ is a measure of fit between a data point $\mathbf{x}$ and a descriptor $\mathbf{t}$, then the M-estimator for $\mathbf{t}$ is the value of $\mathbf{t}$ that minimizes $\sum_{\mathbf{x}} r(\mathbf{x}, \mathbf{t})$. The discussion of the previous section provides measures of fit for M-estimation of location and shape.

Some estimators are not robust, in that their values are sensitive to a few data points that may not represent the information in the bulk of the data. The mean, for example, is sensitive to outliers. The median, on the other hand, is robust to the presence of outliers. The median is also an M-estimator, in that, it is the value of m that minimizes $\sum_x |x - m|$. The measure of fit used for the median is the composition with the measure used for the mean and a concave, increasing function. In particular, $|x - m| = \sqrt{|x - m|^2}$. In other words, we have $f(r) = \sqrt{r}$ with $r = |x - m|^2$. Therefore, the median should be less sensitive than the mean to outliers because the concavity of the square root reduces their influence.

This observation is easily generalized to provide a structure for *robustizing* M-estimators. For a function $r(\mathbf{x}, \mathbf{t})$ measuring fit between $\mathbf{x}$ and $\mathbf{t}$ and a concave, increasing function h, let $\rho(\mathbf{x}, \mathbf{t}) = h(r(\mathbf{x}, \mathbf{t}))$. Since h is increasing, ρ is also a measure of fit with the influence of extreme incompatibility reduced by the concavity of h. Therefore, minimizing $\sum_{\mathbf{x}} \rho(\mathbf{x}, \mathbf{t}) = \sum_{\mathbf{x}} h(r(\mathbf{x}, \mathbf{t}))$ should lead to more robust estimates of $\mathbf{t}$

Table 1: Robustizers.

	$h(r)$	τ
None	r	0
Median	$r^{\frac{1}{2}}$	1
Huber	$\begin{cases} r \\ 2r^{\frac{1}{2}} - 1 \end{cases}$	0.553
Biweight	$\begin{cases} \frac{1}{3}(1-(1-r)^3) \\ \frac{1}{3} \end{cases}$	0.046
Cauchy for $\beta \geq 0$	$\begin{cases} \frac{1}{1-\beta}((1+r)^{1-\beta}-1), \beta \neq 1 \\ \log(1+r), \quad \beta = 1 \end{cases}$	0.176
Fair	$2(r^{\frac{1}{2}} - \log(1+r^{\frac{1}{2}}))$	0.510
Logistic	$2\log(\cosh(r^{\frac{1}{2}}))$	0.689
Talwar	$\begin{cases} r \\ 1 \end{cases}$	0.128
Welsch	$1 - e^{-r}$	0.112
Andrews	$\begin{cases} \frac{2}{\pi^2}(1-\cos(\pi r^{\frac{1}{2}})) \\ \frac{4}{\pi^2} \end{cases}$	0.558

For functions with two descriptions, except for the Cauchy, the first expression applies for $0 \leq r \leq 1$ and the second for $r > 1$. Tuning constants τ are discussed in Section 4.

than would be obtained by minimizing $\sum_{\mathbf{x}} r(\mathbf{x}, \mathbf{t})$.

There are many robust M-estimators of location in the literature. What is not so widely known is that often these can be obtained by composing $|\mathbf{x} - \mathbf{m}|^2$ with some concave function. A list of the estimators is given in Table 1 along with the concave function h. These concave functions can be used with any measures of fit as was illustrated by Verboon and Heiser (1992) who applied the Huber and biweight functions to squared residuals for orthogonal Procrustes analysis.

One caution that must be observed is that the measure of fit is changed by simply rescaling the data. Doing so artificially changes the point in the data space where the robustizing has a particular level of influence. Tuning constants are used to compensate for data scale, that is, use $\rho(\mathbf{x}, \mathbf{t}) = h(\tau r(\mathbf{x}, \mathbf{t}))/\tau$ for a positive tuning constant τ. Dividing by τ is not necessary, but doing so often results in $\tau = 0$ corresponding to no robustizing. This situation occurs when h is differentiable at zero and $h'(0) = H(0) = 1$. Under these conditions $\rho(\mathbf{x}, \mathbf{t})$ approaches $r(\mathbf{x}, \mathbf{t})$ as τ

approaches zero. In fact, whenever h is differentiable at zero, adjusting h by constants will make $h(0) = 0$ and $H(0) = 1$ without affecting the fit interpretation. The functions in Table 1 satisfy these conditions except for the median. Tuning constants can also compensate somewhat for the different levels concavity in different choices for h. Possible tuning constants for each method are also given in Table 1. These are based on values appearing in Holland and Welsch (1977) for achieving 95% efficiency in estimating location under a standard normal model. For clarity, I will not include tuning constants in what follows, but one can apply them as above, if desired.

5. Cluster Analysis

Cluster analysis attempts to identify substructure in a data set or population by grouping objects together into "clusters" based on their being more similar to each other than to other objects in the study. Objective function methods determine descriptions of clusters that minimize a measure of fit between the descriptions and data describing the objects. The problem is that one needs to identify and describe a subset of the data that forms a cluster in the presence of other data that should not belong to the cluster. We need to find a description of a cluster by identifying concentrations of objects that best fit a common description while reducing the effect of objects sufficiently removed from the description that they should be identified with other clusters. Concavity does precisely this. We will see that the contribution to the total fit of a data point is greatest for the cluster description with which it is most compatible, while the influence of fit for other clusters is reduced or even ignored.

For an individual cluster, we will use $r(\mathbf{x}, \mathbf{t})$ as above to measure fit between a data point $\mathbf{x}$ and a cluster descriptor $\mathbf{t}$, such as location and shape.

Three famous objective function clustering procedures are partitioning methods, mixture analysis, and fuzzy clustering.

- **Partitioning:** For $C_1, \ldots, C_k$ a partition of the data into clusters and descriptors $\mathbf{t}_1, \ldots, \mathbf{t}_k$, one for each cluster,

$$\sum_i \sum_{\mathbf{x} \in C_i} r(\mathbf{x}, \mathbf{t}_i)$$

  is the total fit between the data and the given cluster structure, so that one finds the $C_1, \ldots, C_k$ and $\mathbf{t}_1, \ldots, \mathbf{t}_k$ that minimize the badness-of-fit.

- **Mixture analysis:** This approach assumes that the data are a sample from a population whose probability density is a finite mixture of densities of the form $e^{-r(\mathbf{x},\mathbf{t}_i)}$ and the structure is determined by finding $\mathbf{t}$ and $\mathbf{p} = (p_1, \ldots, p_k)$ to maximize

$$\sum_{\mathbf{x}} \log \left(\sum_i p_i e^{-r(\mathbf{x},\mathbf{t}_i)} \right)$$

  where p_i is the probability of belonging to the i-th subpopulation. In this case, $\mathbf{t}$ and $\mathbf{p}$ are maximum likelihood estimates. Redner and Walker (1984) survey the topic.

- **Fuzzy clustering:** This method is based on the notion that some concepts that appear to define sets really do not. For example, who is in the "set of tall people"? There is no clear cutoff in height that would universally be accepted as the dividing line between tall and not tall. A fuzzy set is a function that assigns to each object a number between zero and one that measures the degree to which the object has the property that the set represents. A fuzzy partition of R^d is a vector of k such functions defined on R^d, $\mathbf{u} = (u_1, \ldots, u_k)$, with u_i the membership function for the i-th cluster and restricted so that $\sum_i u_i(\mathbf{x}) = 1$ for each $\mathbf{x}$. A fuzzy clustering is obtained by finding $\mathbf{u}$ and $\mathbf{t}$ to minimize

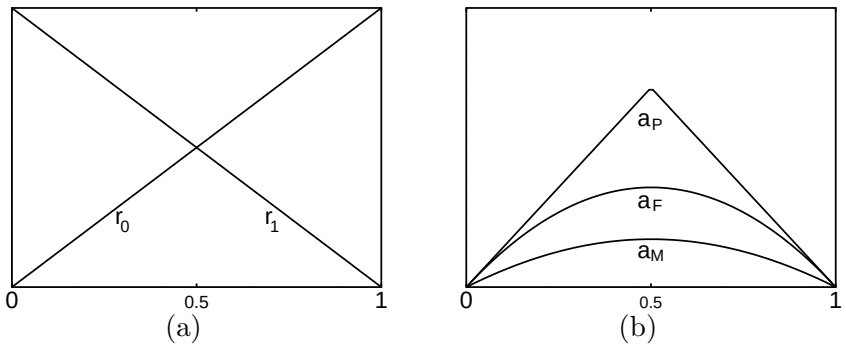
$$\sum_{\mathbf{x}} \sum_i u_i^m(\mathbf{x})\, r(\mathbf{x}, \mathbf{t}_i)$$

  where $m > 1$ is a tuning constant used to adjust the "fuzziness" of the clusters. The larger m is the fuzzier the clusters will be and the closer m is to one, the closer the results are to partitioning. Bezdek and Pal (1991) survey the literature on the subject.

These three objective functions can be converted into compositions of the measures fit for each of the clusters given by $\mathbf{r}(\mathbf{x}, \mathbf{t}) = (r(\mathbf{x}, \mathbf{t}_1), \ldots, r(\mathbf{x}, \mathbf{t}_k))$ with concave functions, increasing in each variable. These functions are obtained for partitioning and fuzzy clustering, by replacing the clusters with the sets $C_1, \ldots, C_k$ and membership functions $u_1, \ldots, u_k$ that minimize the respective objective functions for given descriptors, namely $C_i = \{\mathbf{x} : r(\mathbf{x}, \mathbf{t}_i) = \min_j r(\mathbf{x}, \mathbf{t}_j)\}$ and $u_i(\mathbf{x}) = r(\mathbf{x}, \mathbf{t}_i)^{1/(1-m)} / \sum_j r(\mathbf{x}, \mathbf{t}_j)^{1/(1-m)}$. The conversion for mixtures just puts a minus sign in front to change maximizing to minimizing. The function $r_i^*(\mathbf{x}, \mathbf{t}, \mathbf{p}) = r(\mathbf{x}, \mathbf{t}_i) - \log p_i$ is used to measure fit for mixtures. In fact, the same can be done for partitioning and fuzzy clustering, so that all

Table 2: Concave functions in cluster analysis.

Method	a	A_i
Partitioning	$\min_i(r_i)$	$\begin{cases} 1 \text{ if } r_i = \min_j(r_j) \\ 0 \text{ otherwise} \end{cases}$
Mixture	$-\log(\sum_i e^{-r_i})$	$\frac{e^{-r_i}}{\sum_j e^{-r_j}}$
Fuzzy	$(\sum_i r_i^{1/(1-m)})^{1-m}$	$\left(\frac{r_i^{1/(1-m)}}{\sum_j r_j^{1/(1-m)}}\right)^m$



(a) (b)

Figure 2: Concavity in cluster analysis.

three contain a parameter $\mathbf{p} = (p_1, \ldots, p_k)$ satisfying $\sum_i p_i = 1$ and describing how the data are apportioned among the clusters. The concave functions, a, are shown in Table 2.

The resulting fit functions are

- Partitioning: $\sum_{\mathbf{x}} \min_i r_i^*(\mathbf{x}, \mathbf{t}, \mathbf{p})$
- Mixture: $-\sum_{\mathbf{x}} \log\left(\sum_i e^{-r_i^*(\mathbf{x},\mathbf{t},\mathbf{p})}\right)$
- Fuzzy clustering: $\sum_{\mathbf{x}} \left(\sum_i r_i^*(\mathbf{x}, \mathbf{t}, \mathbf{p})^{1/1-m}\right)^{1-m}$

A simplified illustration will show how concavity facilitates cluster analysis. In Figure 2(a) the line r_0 represents a measure of incompatibility with 0 in that the further x is from 0 the greater the measure. Similarly, r_1 represents a measure of incompatibility with 1. Figure 2(b)

shows the associated functions from Table 2 applied to r_0 and r_1, namely, for partitioning $a_P = \min(r_0, r_1)$, for mixtures $a_M = -\log(e^{-r_0} + e^{-r_1})$ translated by a constant for the purpose of the illustration so that its graph goes through goes through 0 and 1, and for fuzzy clustering $a_F = (r_0^{-1} + r_1^{-1})^{-1}$, using $m = 2$. In terms of incompatibility, a_P ignores incompatibility with 1 for x closer to 0 than to 1 and ignores incompatibility with 0 for x closer to 1 than to 0. The concavity ensures that near 0 the sense of incompatibility is preserved, that is incompatibility with 0 increases as you move away from 0 and similarly for near 1. The functions for mixture and fuzzy clustering are simply smoother versions of the same phenomenon.

One disadvantage to this approach is that the sense of "cluster" is lost. The parameters $\mathbf{t}_i$ describe the clusters in terms of characteristics like location and shape, but what are the clusters as sets of objects? Fortunately, the notion of cluster membership is still present in the A_i's. In particular we have the following interpretations.

- Partitioning:
  $C_i = \{\mathbf{x} : r_i^*(\mathbf{x}, \mathbf{t}, \mathbf{p}) = \min_j r_j^*(\mathbf{x}, \mathbf{t}, \mathbf{p})\} = \{\mathbf{x} : A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}, \mathbf{p})) = 1\}$ is the subset of the data most compatible with the i-th cluster description, that is C_i is the i-th cluster.

- Mixtures:
  $A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}, \mathbf{p})) = p_i e^{-r(\mathbf{x}, \mathbf{t}_i)} / \sum_j p_j e^{-r(\mathbf{x}, \mathbf{t}_j)}$ is the estimate for the conditional probability of belonging to the i-th subpopulation knowing the data $\mathbf{x}$.

- Fuzzy clustering:
  The fuzzy membership function for the ith fuzzy cluster is

$$\begin{aligned} u_i(\mathbf{x}) &= A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}, \mathbf{p}))^{\frac{1}{m}} \\ &= r_i^*(\mathbf{x}, \mathbf{t}, \mathbf{p})^{1/(1-m)} / \sum_j r_j^*(\mathbf{x}, \mathbf{t}, \mathbf{p})^{1/(1-m)}. \end{aligned}$$

The clustering procedures I have discussed so far use continuously varying parameters to describe the clusters. Many procedures do not. For example, the *medoid* method described in Kaufman and Rousseeuw (1990) attempts to identify objects among those under study to serve as descriptors for the location of the clusters, the "medoids". The cluster associated to a given medoid consists of the objects the least dissimilar to it. The medoids are chosen to be the objects for which the sum of the dissimilarities to them of the objects associated to them is the least. We can formulate a badness-of-fit function whose minimum identifies

the medoids and their clusters as follows. Let $\{j_1, \ldots, j_k\}$ be the indices of k distinct objects that are candidates for medoids, and d_{jl} denote the dissimilarity between the j-th and l-th objects, then the badness-of-fit function is

$$\min_{\{j_1,\ldots,j_k\}} \sum_l \min_i d_{j_i l}$$

The concave function $\min(\cdot)$ plays its usual role. In particular, $\min_i$ ignores objects not associated to the i-th medoid and $\min_{\{j_1,\ldots,j_k\}}$ ignores objects not chosen to be medoids. Windham (1985) describes a fuzzy version of the same idea. The reader is encouraged to create a "mixture" version as well.

6. Robust Cluster Analysis

Robust cluster analysis is now quite straight forward to achieve. Simply replace $r(\mathbf{x}, \mathbf{t}_i)$ by $h(r(\mathbf{x}, \mathbf{t}_i))$. This approach was introduced for mixture analysis in Windham (2000), but the concavity in both cluster analysis and M-estimation makes it possible to use the same idea in partitioning and fuzzy clustering. In particular, for any function a from Table 2, the badness-of-fit function

$$\sum_{\mathbf{x}} a(r(\mathbf{x}, \mathbf{t}_1) - \log p_1, \ldots, r(\mathbf{x}, \mathbf{t}_k) - \log p_k) = \sum_{\mathbf{x}} a(\mathbf{r}^*(\mathbf{x}, \mathbf{t}, \mathbf{p}))$$

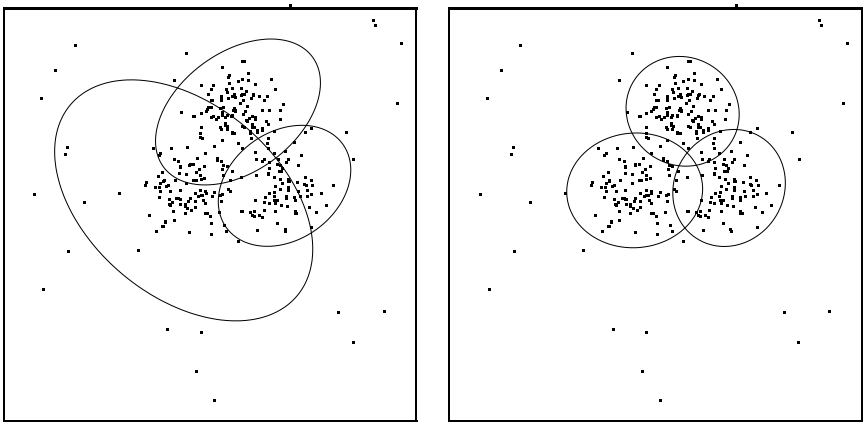
becomes

$$\sum_{\mathbf{x}} a(h(r(\mathbf{x}, \mathbf{t}_1)) - \log p_1, \ldots, h(r(\mathbf{x}, \mathbf{t}_k)) - \log p_k) = \sum_{\mathbf{x}} a(\boldsymbol{\rho}^*(\mathbf{x}, \mathbf{t}, \mathbf{p}))$$

Figure 3 shows the impact of robustizing in an example presented in Windham (2000). Figure 3(a) shows the results of finding three clusters using a mixture clustering function with a Cauchy measure of fit, based on $g(s) = \log(1+s)$. The confusion caused by the presence of the outliers is apparent. Figure 3(b) shows the results obtained with the mixture clustering, Cauchy fit, and the Welsch robustizer with tuning constant 0.112. The robustized result clearly identifies the structure of the bulk of the data without being unduly influenced by the outliers.

7. Minimization Procedures

The role that concavity plays in constructing minimizing algorithms has been recognized for some time. A comprehensive discussion is given in Heiser (1995) where it is viewed as a special case of iterative majorization. It is also described in Lange, Hunter and Yang (2000), which



(a) (b)

Figure 3: Cluster analysis with outliers.

includes the algorithm for finding the median as an example. Both of these articles provide a wealth of references to many other applications of the idea in constructing algorithms.

The iterative majorization with concave functions is based on the following consequence of concavity. From (1), one can see that if you are at $\mathbf{r}_0$ then you can decrease $f(\mathbf{r})$ by decreasing $\mathbf{F}(\mathbf{r}_0)\mathbf{r}$. In particular, if

$$\mathbf{F}(\mathbf{r}_0)\mathbf{r} \leq \mathbf{F}(\mathbf{r}_0)\mathbf{r}_0, \tag{3}$$

then

$$f(\mathbf{r}) \leq \mathbf{F}(\mathbf{r}_0)(\mathbf{r} - \mathbf{r}_0) + f(\mathbf{r}_0) \leq f(\mathbf{r}_0).$$

Iteratively applying (3) will produce a decreasing sequence of values of f. If f is bounded below, the sequence of function values, $f(\mathbf{r})$, will converge. Finally, if f is also increasing in each variable, then $F_i(\mathbf{r}_0) \geq 0$, so $\mathbf{F}(\mathbf{r}_0)\mathbf{r}$ can be decreased by simply decreasing each variable, r_i, and decreasing $f(\mathbf{r})$ has then been reduced to decreasing a weighted version of $\mathbf{r}$. Decreasing $\mathbf{r}$ may not be difficult. In fact, in applications $\mathbf{r}$ is often a function of some parameter important to the context and we are seeking the value of the parameter that minimizes $f(\mathbf{r})$. The function $\mathbf{F}(\mathbf{r}_0)\mathbf{r}$ is often, at least locally, a convex function of that parameter and can be easily minimized.

The articles mentioned above also contain detailed discussions of the convergence properties of the algorithms. I mentioned that when the objective function is bounded below a decreasing sequence of iterates of the function itself will converge. More important is the question of whether the corresponding sequence of parameters converges. In fact,

there is no guarantee that they do converge and even when they do they may not converge to minimizer. These articles along with Bezdek, Hathaway, Howard, Wilson and Windham (1987) and Windham (1987) discuss this problem.

The following describes the iterative procedures for robust estimation, cluster analysis, and finally robust cluster analysis. Each of them is the implementation of (3). The function $\mathbf{F}$ becomes a function H, such as the derivative of an h from Table 1 in robust estimation, and $\mathbf{A} = (A_1, \ldots, A_k)$ from Table 2 in cluster analysis.

Robust Estimation: Robust estimators can be computed from the iterative process based on (3) simply by using the appropriate choice of H, the derivative of a concave, increasing function, in (4) below. The iterations proceed from the current value of $\mathbf{t}$, $\mathbf{t}^c$, to next, $\mathbf{t}^+$. We will use $\arg\min_{\mathbf{t}} l(\mathbf{t})$ to denote the value of $\mathbf{t}$ that minimizes $l(\mathbf{t})$. Beginning with the initial $\mathbf{t}^c$ to be the value of $\mathbf{t}$ that minimizes $\sum_{\mathbf{x}} r(\mathbf{x}, \mathbf{t})$, the iteration is

$$\mathbf{t}^+ = \arg\min_{\mathbf{t}} \sum_{\mathbf{x}} H(r(\mathbf{x}, \mathbf{t}^c)) r(\mathbf{x}, \mathbf{t}) \tag{4}$$

The algorithm reduces the problem of minimizing the robustized criterion to iteratively minimizing a weighted version of the unrobustized function, a problem that is usually easy to solve. For example, if $\mathbf{t} = (\mathbf{m}, \mathbf{S})$ and $r(\mathbf{x}, \mathbf{t}) = (\mathbf{x} - \mathbf{m})^T \mathbf{G}(\mathbf{S})(\mathbf{x} - \mathbf{m}) + \mathrm{tr}(\mathbf{g}(\mathbf{S}) - \mathbf{G}(\mathbf{S})\mathbf{S})$ as in Section 3, then the parameter updates are as follows. Letting $w^c(\mathbf{x}) = H(r(\mathbf{x}, \mathbf{t}^c)) / \sum_{\mathbf{x}} H(r(\mathbf{x}, \mathbf{t}^c))$, we have

$$\begin{aligned} \mathbf{m}^+ &= \sum_{\mathbf{x}} w^c(\mathbf{x}) \mathbf{x} \\ \mathbf{S}^+ &= \sum_{\mathbf{x}} w^c(\mathbf{x}) (\mathbf{x} - \mathbf{m}^+)(\mathbf{x} - \mathbf{m}^+)^T. \end{aligned}$$

The value $w^c(\mathbf{x})$ simply weights the contribution of $\mathbf{x}$ in accordance with the effect of the concave function h used for robustness.

Clustering: The algorithm for minimizing the objective functions iterates from $\mathbf{t}^c, \mathbf{p}^c$ to $\mathbf{t}^+, \mathbf{p}^+$ by

$$\begin{aligned} \mathbf{t}_i^+ &= \arg\min_{\mathbf{t}} \sum_{\mathbf{x}} A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}^c, \mathbf{p}^c)) r(\mathbf{x}, \mathbf{t}) \\ p_i^+ &= \arg\max_{p_i} \sum_{\mathbf{x}} A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}^+, \mathbf{p}^c)) \log p_i \\ &= \frac{\sum_{\mathbf{x}} A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}^+, \mathbf{p}^c))}{\sum_j \sum_{\mathbf{x}} A_j(\mathbf{r}^*(\mathbf{x}, \mathbf{t}^+, \mathbf{p}^c))}. \end{aligned}$$

In the case where the parameters are location and scale, that is $\mathbf{t}_i = (\mathbf{m}_i, \mathbf{S}_i)$, the updates for those parameters are

$$\begin{aligned} \mathbf{m}_i^+ &= \sum_{\mathbf{x}} w_i^c(\mathbf{x})\mathbf{x} \\ \mathbf{S}_i^+ &= \sum_{\mathbf{x}} w_i^c(\mathbf{x})(\mathbf{x}-\mathbf{m}^+)(\mathbf{x}-\mathbf{m}^+)^T \end{aligned}$$

where $w_i^c(\mathbf{x}) = A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}^c, \mathbf{p}^c))/\sum_{\mathbf{x}} A_i(\mathbf{r}^*(\mathbf{x}, \mathbf{t}^c, \mathbf{p}^c))$ weights the contribution of $\mathbf{x}$ to the next parameter estimates for the i-th cluster according to its "membership" in the i-th cluster as described by the current parameter estimates.

Robust Clustering:

$$\begin{aligned} \mathbf{t}_i^+ &= \arg\min_{\mathbf{t}} \sum_{\mathbf{x}} A_i(\boldsymbol{\rho}^*(\mathbf{x}, \mathbf{t}^c, \mathbf{p}^c))H(r(\mathbf{x}, \mathbf{t}_i^c))r(\mathbf{x}, \mathbf{t}) \\ p_i^+ &= \frac{\sum_{\mathbf{x}} A_i(\boldsymbol{\rho}^*(\mathbf{x}, \mathbf{t}^+, \mathbf{p}^c))}{\sum_j \sum_{\mathbf{x}} A_j(\boldsymbol{\rho}^*(\mathbf{x}, \mathbf{t}^+, \mathbf{p}^c))} \end{aligned}$$

When the parameters are location and scale, the updates for those parameters are

$$\begin{aligned} \mathbf{m}_i^+ &= \sum_{\mathbf{x}} w_i^c(\mathbf{x})\mathbf{x} \\ \mathbf{S}_i^+ &= \sum_{\mathbf{x}} w_i^c(\mathbf{x})(\mathbf{x}-\mathbf{m}^+)(\mathbf{x}-\mathbf{m}^+)^T \end{aligned}$$

where the weight

$$w_i^c(\mathbf{x}) = A_i(\boldsymbol{\rho}^*(\mathbf{x}, \mathbf{t}^c, \mathbf{p}^c))H(r(\mathbf{x}, \mathbf{t}_i^c))/\sum_{\mathbf{x}} A_i(\boldsymbol{\rho}^*(\mathbf{x}, \mathbf{t}^c, \mathbf{p}^c))H(r(\mathbf{x}, \mathbf{t}_i^c))$$

has two terms, one for clustering and the other for robustness.

8. Conclusion

The influence of extreme values on parameter estimation in data analysis can be controlled with concave functions without seriously complicating the problem to be solved. This paper illustrates this fact with three examples where different things need to be controlled. In the first, extreme values of scale were controlled so that well-behaved measures of fit could be built that would have the covariance or scatter matrix as

a minimizer. In the M-estimation example, the influence of outliers on parameter estimates was controlled with concave functions. Finally, in finding a description of a cluster in a cluster analysis, the influence of data from other clusters is reduced by combining fits to cluster descriptions with concave functions. Moreover, the common thread of applying concave functions to control extreme values allows one to combine all three to build a procedure for robust cluster analysis, that estimates location and scale within a cluster.

References

BEZDEK, J.C., HATHAWAY, R.J., HOWARD, R.E., WILSON, C.A. and WINDHAM, M.P. (1987), "Local convergence analysis of a grouped variable version of coordinate descent," *Journal of Optimization Theory and Applications, 54,* 471–477.

BEZDEK, J.C. and PAL, S.K. (1991), *Fuzzy Models for Pattern Recognition,* New York: IEEE Press.

HAMPEL, F.R., RONCHETTI, E.M., ROUSSEEUW, P.J. and STAHEL, W.A. (1986), *Robust Statistics: The Approach Based on Influence Functions,* New York: Wiley.

HEISER, W.J. (1995), "Convergent computation by iterative majorization: theory and applications in multidimensional data analysis," *Recent Advances in Descriptive Multivariate Analysis,* Ed. W. Krzanowski, Oxford: Clarendon Press, 157–189.

HOLLAND, P.W. and WELSCH, R.E. (1977), "Robust regression using iteratively reweighted least-squares," *Communications in Statistics Theory and Methods, A6(9),* 813-827.

HUBER, P.J. (1981), *Robust Statistics,* New York: Wiley.

KAUFMAN, L. and Rousseeuw, P. (1990), *Finding Groups in Data: An Introduction to Cluster Analysis,* New York: Wiley.

LANGE, K., HUNTER, D. and YANG, I. (2000), "Optimization transfer using surrogate objective functions," *Journal of Computational and Graphical Statistics, 9,* 1–59.

REDNER, R.A. and WALKER, H.F. (1984), "Mixture densities, maximum likelihood and the EM algorithm," *SIAM Review, 26,* 195–239.

VERBOON, P. and HEISER, W.J. (1992), "Resistant orthogonal Procrustes analysis," *Journal of Classification, 9,* 237–256.

WINDHAM, M.P. (1985), "Numerical classification of proximity data with assignment measures," *Journal of Classification, 2,* 157–172.

WINDHAM, M.P. (1987), "Parameter modification for clustering criteria," *Journal of Classification, 4,* 191–214.

WINDHAM, M.P. (2000), "Robust clustering," *Data Analysis: Scientific Modeling and Practical Application,* Eds. W. Gaul, O. Opitz and M. Schader, Berlin: Springer, 385–392.