CHAPTER 6

Partition by Exact Optimization

6.1 INTRODUCTION

A partition of a set of objects is a family of subsets such that each object lies in exactly one member of the family. If $R(M, K)$ is the number of partitions of M objects into K sets, it can be seen by induction on M that

$$R(M, K) = KR(M - 1, K) + R(M - 1, K - 1).$$

Also $R(M, 1) = R(M, M) = 1$. As a result, the number of partitions increases approximately as K^M; for example, $R(10, 3) = 9330$.

Optimization techniques in clustering associate an error function $e[P(M, K)]$ with each partition $P(M, K)$ of M objects into K clusters and seek that partition which minimizes $e[P(M, K)]$. Because of the very large number of possible partitions it is usually impractical to do a complete search, so it is necessary to be satisfied with a "good" partition rather than a best one. Sometimes prior constraints or mathematical consequences of the error function e reduce the number of possible partitions, and a complete search to find the exact optimum is possible.

Such a case occurs when the data points are ordered, for example, by time, as in Table 6.1, which reports the winning times in the 100-m run in the Olympics from 1896 to 1964. The data points are ordered by the year of the Olympics. It is sensible to require each cluster to correspond to an interval of years. The partition will reveal whether or not there were periods of years when the winning times remained fairly constant. The number of possible partitions into K clusters is now $O(M^K)$ rather than $O(K^M)$, and even further reduction is possible when an error criterion is used that is additive over clusters.

6.2 FISHER ALGORITHM

Preliminaries. This algorithm is due to Fisher (1958). Objects are labeled $1, 2, \ldots, M$, and clusters are constrained to consist of intervals of objects $(I, I + 1, I + 2, \ldots, J - 1, J)$. There are only $\frac{1}{2}M(M + 1)$ possible clusters. There is a diameter $D(I, J)$ associated with the cluster $(I, I + 1, \ldots, J)$ such that the error of a partition $P(M, K)$ into K clusters $(I_1 = 1, I_1 + 1, \ldots, I_2 - 1)$, $(I_2, I_2 + 1, \ldots, I_3 - 1), \ldots, (I_{K-1}, \ldots, I_K - 1)$, $(I_K, I_K + 1, \ldots, M)$ is

$$e[P(M, K)] = \sum \{1 \leq J \leq K\}\, D(I_J, I_{J+1} - 1).$$

The error of a partition is thus the sum of cluster diameters over the clusters it contains.

Table 6.1 Olympic Track 1896–1964

From *The World Almanac* (1966), New York World-Telegram, New York.

In tenths of seconds (- denotes missing)

	100M	200M	400M	800M	1500M	5000M	10000M
1896	120	-	542	1310	2732	-	-
1900	108	222	494	1214	2460	-	-
1904	110	216	492	1160	2454	-	-
1906	112	-	532	1212	2520	-	-
1908	108	224	500	1128	2434	-	-
1912	108	217	482	1119	2368	8766	18808
1920	108	220	496	1134	2418	8956	19058
1924	106	216	476	1124	2336	8712	18232
1928	108	218	478	1118	2332	8780	18188
1932	103	212	462	1098	2312	8700	18114
1936	103	207	465	1129	2278	8622	18154
1948	103	211	462	1092	2298	8576	17996
1952	104	207	459	1092	2252	8460	17570
1956	105	206	467	1077	2212	8196	17256
1960	102	205	449	1063	2156	8234	17122
1964	100	203	451	1051	2181	8288	17044

The spring of this algorithm is the relation between optimum partitions into K clusters and optimum partitions into $K-1$ clusters. Let $\mathbf{P}(I, L)$ denote the optimum partition of objects $1, 2, \ldots, I$ into L clusters for $I \leq M$, $L \leq K$. Suppose that $\mathbf{P}(M, K) = (I_1, \ldots, I_2 - 1), (I_2, \ldots, I_3 - 1), \ldots, (I_K, \ldots, M)$. Then necessarily $\mathbf{P}(I_K - 1, K - 1) = (I_1, \ldots, I_2 - 1)(I_2, \ldots, I_3 - 1) \cdots (I_{K-1}, \ldots, I_K - 1)$. Since error is additive, if this were not true, $e[\mathbf{P}(M, K)]$ could be reduced by varying

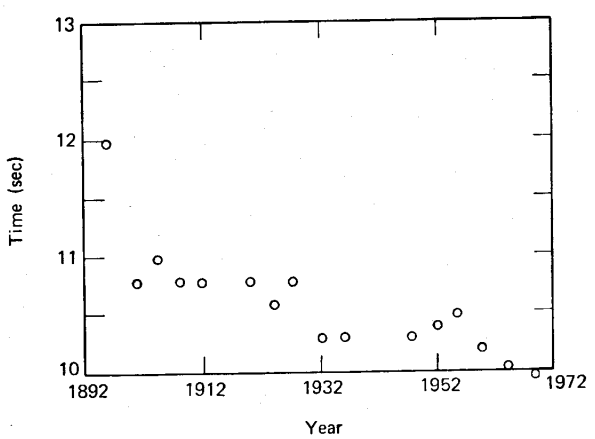

Figure 6.1 Times in the 100-m run, Olympic games.

$I_2, I_3, \ldots, I_{K-1}$. Using this relationship, the algorithm proceeds by successively computing optimal partitions into 2, 3, 4, ..., K clusters, building on the $(K-1)$-partitions to find the K-partitions. This is a "dynamic programming" procedure (Bellman and Dreyfus, 1962).

STEP 1. Compute the diameter $D(I, J)$ for the cluster $(I, I+1, \ldots, J)$, for all I, J such that $1 \leq I < J \leq M$.

STEP 2. Compute the errors of the optimal partitions, $2 \leq I \leq M$, by $e[\mathbf{P}(I, 2)] = \min [D(1, J-1) + D(J, I)]$ over the range $2 \leq J \leq I$.

STEP 3. For each L $(3 \leq L \leq K)$ compute the errors of the optimal partitions $e[\mathbf{P}(I, L)]$ $(L \leq I \leq M)$ by

$$e[\mathbf{P}(I, L)] = \min \{e[\mathbf{P}(J-1, L-1)] + D(J, I)\}$$

over the range $L \leq J \leq I$.

STEP 4. The optimal partition $\mathbf{P}(M, K)$ is discovered from the table of errors $e[\mathbf{P}(I, L)]$ $(1 \leq L \leq K, 1 \leq I \leq M)$ by first finding J so that

$$e[\mathbf{P}(M, K)] = e[\mathbf{P}(J-1, K-1)] + D(J, M).$$

The last cluster is then $(J, J+1, \ldots, M)$. Now find J^* so that $e[\mathbf{P}(J-1, K)] = e[\mathbf{P}(J-1, K-1)] + D(J^*, J-1)$. The second-to-last cluster of $\mathbf{P}(M, K)$ is $(J^*, J^*+1, \ldots, J-1)$, and so on.

NOTE. The partition is guaranteed optimal, but it is not necessarily unique.

6.3 FISHER ALGORITHM APPLIED TO OLYMPIC TIMES

It is first necessary to define cluster diameter. Let $X(I)$ be the value associated with the Ith object. A standard measure of diameter of the cluster $I, I+1, \ldots, J$ is

$$D(I, J) = \sum \{I \leq L \leq J\} [X(L) - \bar{X}]^2,$$

where

$$\bar{X} = \sum \{I \leq L \leq J\} \frac{X(L)}{J - I + 1}$$

is the mean of the values in the cluster.

Another measure, more convenient for hand calculation, is

$$D(I, J) = \sum \{I \leq L \leq J\} |X(L) - \tilde{X}|,$$

where the median $\tilde{X}$ is both no greater than and no less than half the values $X(L)$ $(I \leq L \leq J)$.

STEP 1. Compute the diameter of all clusters. With 16 objects there are $16 \times \frac{17}{2} = 126$ diameters. For example, $D(7, 12)$ is computed from the times 10.6, 10.8, 10.3, 10.3, and 10.4, which have median 10.4. The deviations are 0.2, 0.4, 0.1, 0.1, 0 which have a sum of 0.8, so $D(7, 12) = 0.8$. All diameters are given in Table 6.2.

Table 6.2 Diameters of Clusters of Olympic Times (in Tenths of Seconds)

Diameter in the interval (I, J) is the sum of absolute deviations from the median of observations $X(I), \ldots, X(J)$.

	J	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
I	1	0	12	12	14	14	14	16	16	21	26	31	35	38	44	50	57
	2		0	2	2	2	2	4	4	9	14	19	23	26	30	36	42
	3			0	2	2	2	4	4	9	14	19	23	24	28	33	39
	4				0	0	0	2	2	7	12	17	19	20	23	28	33
	5					0	0	2	2	7	12	15	17	17	20	24	29
	6						0	2	2	7	10	13	13	14	16	20	24
	7							0	2	5	8	8	9	10	12	15	16
	8								0	5	5	5	6	8	9	12	16
	9									0	0	0	1	3	4	7	11
	10										0	0	1	3	4	7	11
	11											0	1	2	4	7	11
	12												0	1	3	7	11
	13													0	3	5	8
	14														0	2	3
	15															0	1
	16																0

STEP 2. All optimal 2-partitions $\mathbf{P}(I, 2)$ are to be computed. It is necessary to remember only $e[\mathbf{P}(I, 2)]$ for later steps. As an example, $e[\mathbf{P}(4, 2)]$ is the minimum of

$$D(1, 3) + D(4, 4) = 12,$$

$$D(1, 2) + D(3, 4) = 12 + 2 = 14,$$

and

$$D(1, 1) + D(2, 4) = 2.$$

Thus $e[\mathbf{P}(4, 2)] = 2$. All the errors for optimal partitions are given in Table 6.3.

STEP 3. The optimal 3-partitions are developed from the optimum 2-partitions. For example, $e[\mathbf{P}(6.3)]$ is the minimum of

$$e[\mathbf{P}(5, 2)] + D(6, 6) = 2 + 0 = 2,$$

$$e[\mathbf{P}(4, 2)] + D(5, 6) = 2 + 0 = 2,$$

$$e[\mathbf{P}(3, 2)] + D(4, 6) = 2 + 0 = 2,$$

and

$$e[\mathbf{P}(2, 2)] + D(3, 6) = 0 + 2 = 2,$$

so $e[\mathbf{P}(6, 3)] = 2$.

Similarly, the optimal 4-partitions are developed from the optimum 3-partitions.

Table 6.3 Errors of Optimal Partitions

$e[\mathbf{P}(I, L)]$ is the error of optimal partition of objects 1, 2, . . . , I into L clusters.

I	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
	0	12	12	14	14	14	16	16	21	26	31	35	38	44	50	57
		0	2	2	2	2	4	4	9	14	16	17	19	20	23	27
			0	2	2	2	2	4	4	4	4	5	7	8	11	15
				0	0	0	2	2	4	4	4	4	5	7	8	9
					0	0	0	2	2	2	2	3	4	5	7	8
						0	0	0	2	2	2	2	3	4	5	6
							0	0	0	0	0	1	2	3	4	5
								0	0	0	0	0	1	2	3	4
									0	0	0	0	0	1	2	3
										0	0	0	0	0	1	2
											0	0	0	0	0	1
												0	0	0	0	0
													0	0	0	0
														0	0	0
															0	0
																0

STEP 4. To find the optimal partition of 16 into, say four clusters, first find J such that

$$e[\mathbf{P}(16, 4)] = e[\mathbf{P}(J - 1, 3)] + D(J, 16).$$

Such a J is $J = 15$. The last cluster is (15, 16). Now find J so that

$$e[\mathbf{P}(14, 3)] = e[\mathbf{P}(J - 1, 2)] + D(J, 14).$$

Thus $J = 9$, and the second last cluster is (9, 14).

Since $e[\mathbf{P}(8, 2)] = D(1, 1) + D(2, 8)$, the first two clusters are (1) and (2, 8).

Thus, $\mathbf{P}(16, 4) = (1), (2, \ldots, 8), (9, \ldots, 14), (15, 16)$. In terms of the observations,

$\mathbf{P}(16, 4)$

= (12) (10.8 11 10.8 10.8 10.8 10.6 10.8) (10.3 10.3 10.3 10.4 10.5 10.2) (10.0 9.9).

This seems a reasonable partition of the data. Some idea of the best number of clusters may be obtained by plotting $e[\mathbf{P}(16, K)]$ against K, as in Figure 6.2. There are sharp decreases in error at $K = 2$ and $K = 3$, a noticeable decrease at $K = 4$, and trivial decreases for larger K. The correct number of clusters is 3 or 4.

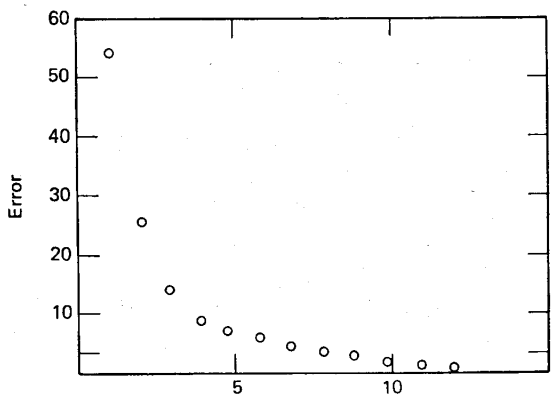

Figure 6.2 Errors of optimal K-partitions of Olympic data, using absolute deviations to measure diameter.

6.4 SIGNIFICANCE TESTING AND STOPPING RULES

Let $X(1), X(2), \ldots, X(M)$ denote the M observations in order, and consider a statistical model for a partition into K clusters:

$$(I_1 = 1, I_1 + 1, \ldots, I_2 - 1)(I_2, I_2 + 1, \ldots, I_3), \ldots, (I_K, I_K + 1, \ldots, M),$$

for which $X(1), X(2), \ldots, X(I_2 - 1)$ are independent observations from the density $f(X \mid \theta_1)$, $X(I_2), X(I_2 + 1), \ldots, X(I_3 - 1)$ are independent observations from the density $f(X \mid \theta_2)$, and so on, up to $X(I_K), \ldots, X(M)$ are independent observations from the density $f(X \mid \theta_K)$. (Note that both X and θ may have dimension greater than 1.) The likelihood of the joint parameters $\theta_1, \theta_2, \ldots, \theta_K$ is

$$\prod \{1 \leq L \leq K\} \prod \{I_L \leq J < I_{L+1}\} f[X(J) \mid \theta_L].$$

The maximum log likelihood of the observations is

$$LL = \sum \{1 \leq L \leq K\} \sum \{I_L \leq J < I_{L+1}\} \log f[X(J) \mid \hat{\theta}_L],$$

where $\hat{\theta}_L$ is the maximum likelihood estimate of θ_L, based on observations in the Lth clusters. Note that, if $D(I, J)$, the cluster diameter, is defined by

$$D(I, J) = -\max_{\theta} \sum \{I \leq L \leq J\} f[X(L) \mid \theta],$$

then $-LL$ is the sum of $D(I, J)$'s corresponding to the clusters. Minimizing $-LL$ means that clusters are found, and parameter values within clusters are estimated, to make the given observations most probable.

In this way, the density f, which relates the observations X to the cluster parameter θ, generates a cluster diameter and an additive error measure for partitions. For example, with the double exponential $f(X \mid \theta) = 0.5 \exp(-|X - \theta|)$, the measure of cluster diameter is

$$D(I, J) = \sum \{I \leq L \leq J\} |X(L) - \tilde{X}|,$$

where $\tilde{X}$ is the cluster median. The more common normal density, $f(X, \theta) = \exp(-\frac{1}{2}(X - \theta)^2)/\sqrt{2\pi}$ leads to the familiar sum of squared deviations diameter $D(I, J) = \sum \{I \leq L \leq J\} [X(L) - \bar{X}]^2$, where $\bar{X}$ is the cluster mean.

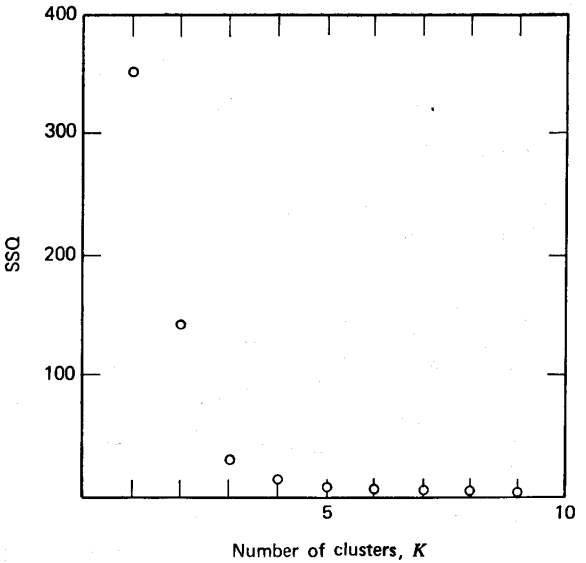

Figure 6.3 Sum of squared errors within optimal partitions of Olympic data.

The optimal partitions of the Olympic data with this criterion are shown in Table 6.4 and Figure 6.3. It will be seen that the principal decreases in the sum of squares come in the second and third partitions, and later decreases are relatively minor. This agrees with the intuition that about three clusters seem right to describe this data.

When are $K + 1$ rather than K clusters necessary? A naive approach supposes that K distinct clusters are present in the data, and that $K + 1$ clusters are obtained by splitting one of these clusters in two arbitrarily. [In reality, one of the clusters would be split optimally if the clusters were really distinct, and in general there will be no simple relation between the optimal K-partition and the optimal $(K + 1)$-partition.] Then the mean square ratio

$$(N - K - 1)\left(\frac{e[P(N, K)]}{e[P(N, K + 1)]} - 1\right)$$

is distributed as $F_{1,N-K-1}$ if the observations are normal. It therefore seems worthwhile to study this ratio of mean square reduction to mean square within clusters, and to take a large value of this quantity to indicate that $(K + 1)$-clusters are necessary. For the Olympic data, the 3-cluster stands out in the mean square ratio table. (The values for $K = 10, 11$ are large but should be ignored; in these cases the mean square error within clusters is grossly underestimated because the data are rounded to tenths of seconds.)

When is the mean square error ratio larger than would be expected from random normal observations? One approach to this is through asymptotic theory as the number of observations becomes large. It is possible to show (by an argument too long to fit in the margin) that the number of objects in the clusters alternates between very large and very small; that is, if $n(L)$ denotes the number of objects in the Lth cluster, $n(L)/n(L + 1)$ is near zero or unity as $M \to \infty$. For example, an optimal partition

Table 6.4 Optimal Partitions of Olympic Data Using Sum of Squared Deviations From Mean as Cluster Diameter

PARTITION SIZE	MSQ RATIO	SUM OF SQUARES	PARTITION (OBSERVATION IN TENTHS OF SECONDS)
1		352.00	120 108 110 108 108 108 106 108 103 103 103 104 105 102 100 99
2	20.7	142.93	120/108 110 108 108 108 106 108 103 103 103 104 105 102 100 99
3	49.0	30.00	120/108 110 108 108 108 106 108/103 103 103 104 105 102 100 99
4	14.0	13.83	120/108 110 108 108 108 106 108/103 103 103 104 105 102/100 99
5	2.8	11.03	120/108 110 108 108 108/106 108/103 103 103 104 105 102/100 99
6	2.4	9.03	120/108 110 108 108 108/106 108/103 103 103/104 105/102 100 99
7	4.1	6.20	120/108 110 108 108 108/106 108/103 103 103/104 105/102/100 99
8	3.8	4.20	120/108 110 108 108 108/106/108/103 103 103/104 105/102/100 99
9	2.8	3.00	120/108 110/108 108 108/106/108/103 103 103/104 105/102/100 99
10	12.0	1.00	120/108/110/108 108 108/106/108/103 103 103/104 105/102/100 99
11	5.0	0.50	120/108/110/108 108 108/106/108/103 103 103/104/105/102/100 99
12	∞	0.00	120/108/110/108 108 108/106/108/103 103 103/104/105/102/100/99

of 500 random normals into 10 clusters yielded cluster sizes 183, 3, 63, 1, 100, 2, 10, 39, 98, 1. Also the mean square error ratio slowly approaches infinity as $M \to \infty$.

Empirical sampling for moderate M shows surprising uniformity in the distributions of the root mean square ratio for various K (Table 6.5). The root mean square ratio has an expectation very close to 2 for a wide range of M and K. The variances depend on M, the total number of objects, but not much on K. The expected values increase slightly with M but are still near 2, even for $M = 500$. The convergence to infinity demanded by asymptotic theory is slow. The various ratios for different K are approximately independent normal variables with the same mean and variance (Figure 6.4). Therefore, under the null hypothesis that no clusters exist, the largest ratio has approximately the distribution of the largest of a number of independent normals. This reference distribution thus provides an approximate significance level for the largest ratio.

6.5 TIME AND SPACE

The Fisher algorithm requires $O(M^2K)$ additions, where M is the number of objects and K is the number of clusters. It is thus feasible for 100–1000 objects. It may be a high price to pay to get exact optimization. During the algorithm it is necessary to store $M \times K$ errors corresponding to optimal partitions $\mathbf{P}(I, L)$ $(1 \leq I \leq M, 1 \leq L \leq K)$.

In Table 6.4, the optimal partitions for various K almost have hierarchical structure; that is, for any two clusters in any two partitions, one cluster includes the other or they are disjoint. The only exception to this rule is the cluster (103 103 103 104 105 102). If it is known that the final partitions have hierarchical structure, a shorter algorithm proceeds as follows.

STEP 1. Split the sequence $1, 2, \ldots, M$ optimally into two clusters by choosing one of M possible split points.

Table 6.5 Empirical Distributions of Root Mean Square Ratio, Based on Random Normal, 500 Trials.

N = 6	K = 2	K = 3							
Expectation	2.154	2.271							
Variance	1.731	1.755							
N = 11	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7			
Expectation	2.030	2.191	1.961	2.010	2.013	1.947			
Variance	.592	.586	.476	.528	.660	.658			
N = 16	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
Expectation	2.020	2.260	1.992	1.968	1.963	1.893	1.932	1.871	1.898
Variance	.529	.487	.359	.287	.268	.243	.361	.256	.374
N = 50	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
Expectation	2.090	2.433	2.144	2.134	2.115	2.003	2.004	1.971	1.969
Variance	.360	.295	.193	.189	.148	.113	.094	.080	.080
(200 TRIALS)									
N = 100	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
Expectation	2.184	2.528	2.246	2.285	2.211	2.179	2.216	2.105	2.108
Variance	.294	.242	.192	.120	.100	.094	.091	.082	.062
(50 TRIALS)									
N = 500	K = 2	K = 3	K = 4	K = 5	K = 6	K = 7	K = 8	K = 9	K = 10
Expectation	2.3	2.9	2.6	2.4	2.5	2.3	2.5	2.4	2.4
(2 TRIALS)									

(N = Number of observations in sequence; K = Number of clusters in partition)

STEP 2. Split optimally into three clusters by choosing one of M possible split points—some in the first cluster from Step 1, the remainder in the second cluster.

STEP 3. Split optimally into four clusters by choosing one of M possible split points, and so on. This procedure requires only $O(MK)$ additions. This algorithm could be used to obtain good (not guaranteed best) partitions of long sequences.

6.6 THINGS TO DO

6.6.1 Running the Fisher Algorithm

This algorithm is appropriate when objects are already ordered by some overriding variable such as time. There is no particular requirement that the data be one-dimensional. Vietnam combat deaths over time (Table 6.6) might be analyzed to detect different phases of U.S. involvement.

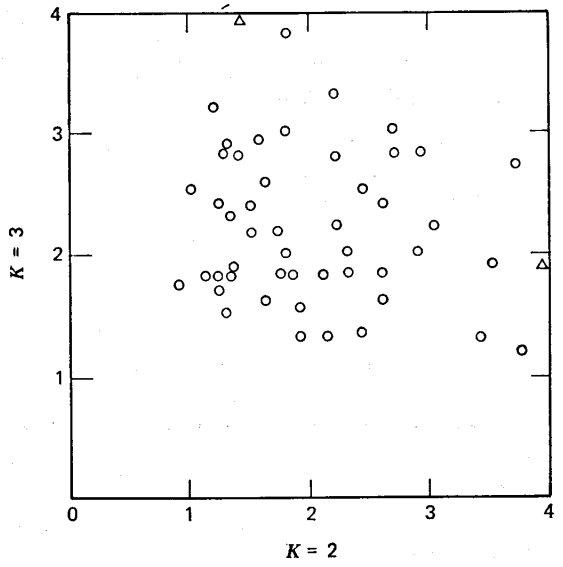

Figure 6.4 Independence of root mean square ratios, computed from 50 normal samples. For 11 observations, ratios for 2- and 3-partitions.

Table 6.6 Combat Deaths in Indochina

		US	SVN	THIRD	ENEMY		US	SVN	THIRD	ENEMY		US	SVN	THIRD	ENEMY
'66	JAN	282	903	74	2648	'68	1202	2905	111	15217	'70	343	1768	69	9187
	FEB	435	1359	58	4727		2124	5025	147	39867		386	1417	36	8828
	MAR	507	1145	59	5685		1543	2570	88	17371		449	1674	75	10335
	APR	316	945	30	2818		1410	1922	85	12215		526	2642	79	13063
	MAY	464	961	19	4239		2169	3467	85	24086		754	2851	58	17256
	JUN	507	1185	41	4815		1146	1974	92	10319		418	2873	63	7861
	JUL	435	1006	32	5297		813	1409	65	6653		332	1711	71	7183
	AUG	396	914	44	5860		1080	2393	73	15478		319	1720	63	6446
	SEP	419	803	30	4459		1053	2164	58	12543		219	1734	46	6138
	OCT	340	844	63	5665		600	1169	70	8168		170	1491	57	5549
	NOV	475	907	87	5447		703	1408	38	9632		167	1619	48	5607
	DEC	432	981	29	3864		749	1509	67	9600		138	1846	39	6185
'67	JAN	520	914	77	6064	'69	795	1664	76	10955	'71	140	1616	30	6155
	FEB	662	885	95	7341		1073	2072	85	14086		221	2435	48	11704
	MAR	944	1297	54	9351		1316	2186	90	19805		272	3676	104	19858
	APR	710	1057	56	6227		847	1710	52	14539		226	2198	86	10457
	MAY	1233	1184	112	9808		1209	2251	92	17443		138	2091	50	9094
	JUN	830	981	74	7354		1100	1867	75	16825		108	1846	44	7648
	JUL	781	676	102	7923		638	1455	64	10237		65	1389	44	6247
	AUG	535	1068	90	5810		795	1625	74	12373		67	1488	32	6165
	SEP	775	1090	149	6354		477	1543	60	10369		78	1607	27	6300
	OCT	733	1066	96	6272		377	1597	80	8747		29	1574	20	5744
	NOV	881	1299	98	7662		446	2105	62	11639		19	1161	14	4283
	DEC	774	1199	102	7938		341	1758	56	9936		17	988	26	4439

From *Unclassified Statistics on Southeast Asia* (1972), Department of Defense, OASD (Comptroller), Directorate for Information Operations.

6.6.2 Real-Valued Data

For clustering a single real variable to minimize the within-cluster sum of squares, the clusters must be convex, which means they must consist of points lying in an interval. The Fisher algorithm may be applied to the ordered points to find the exactly optimal partition into K clusters. This exact partition may be compared with locally optimal partitions obtained by approximate techniques, such as the K-means algorithm.

6.6.3* Estimating Densities

Given that a density is unimodal in an interval, there is a well-known maximum likelihood technique for estimating it (each point in the interval is tried for the mode, and for each modal point the density is first estimated as the reciprocal of the intervals between points and then neighboring intervals are averaged if they violate the monotonicity required by unimodality). Using the Fisher algorithm, maximum likelihood densities with K modes may be computed.

6.6.4 Sequential Splitting

As justification for the hierarchical algorithm in Section 6.5, suppose the data consist of M real values in time and that the time interval is divided into K intervals within each of which the values are constant. The error function is within-cluster sum of squares. There is a K-partition for which the error is zero, and this will be discovered by the hierarchical algorithm.

6.6.5 Updating Sums of Absolute Deviations

The median of a set of numbers $X(1), \ldots, X(M)$ is any number $\tilde{X}$ such that $\tilde{X} \leq X(I)$ occurs at least $M/2$ times and $\tilde{X} \geq X(I)$ occurs at least $M/2$ times. Let $\tilde{X}^*$ be the median (there may be more than one) closest to X. Then the minimum sum of absolute deviations for $X(1), \ldots, X(M), X$ is the minimum sum of absolute deviations for $X(1), \ldots, X(M)$ plus $|X - \tilde{X}^*|$.

REFERENCES

BELLMAN, R. E., and DREYFUS, S. E. (1962). *Applied Dynamic Programming*, Princeton U. P., Princeton N.J. On p. 15, the principle of optimality is stated. An optimal policy has the property that, whatever the initial state and initial decision are, the remaining decisions must constitute an optimal policy with regard to the state resulting from the first decision. A typical formulation of a dynamic programming problem is as follows: Maximize $R(x_1, x_2, \ldots, x_N) = \Sigma g_i(x_i)$ over the region $x_i \geq 0$, $\Sigma x_i = x$. Define $f_N(x)$ to be this maximum. Then

$$f_N(x) = \max_{0 \leq x_N \leq x} [g_N(x_N) + f_{N-1}(x - x_N)].$$

As an example on p. 104, consider the problem of meeting a series of demands in time when it is expensive to have excess capacity or to change the capacity. Thus, if the demands are r_k, the capacities are x_k, the loss due to excess capacity is $\varphi_k(x_k - r_k)$, and the loss due to a change in capacity is $\psi_k(x_k - x_{k-1})$, the problem is to minimize $\sum [\varphi_k(x_k - r_k) + \psi_k(x_k - x_{k-1})]$ subject to $x_k \geq r_k$.

FISHER, W. D. (1958). On grouping for maximum homogeneity. *J. Am. Stat. Assoc.* **53**, 789–798. Given a set of K objects, such that each object has a weight w_i and a numerical measure a_i, Fisher discusses techniques of assigning the objects to G groups so as to minimize $\sum w_i(a_i - \bar{a}_i)^2$, where $\bar{a}_i$ is the weighted mean of a's in the group to which a_i is assigned. He shows that each group must be convex in the a's; that is, each group consists of all the a's in an interval, if the optimization takes place without prior constraints. If the objects are ordered *a priori*, so that clusters are sequences of objects in the original order, he remarks that the additive property of the sum of squares criterion makes it possible to reduce the computation by relating optimum partitions into G clusters to optimum partitions into fewer clusters. He also notes a nonlinear programming formulation of the problem: Let x_{hi} denote the fractional part of a_i that is assigned to group h ($h = 1, 2, \ldots, G$). Define $\bar{a}_h = (\sum_i x_{hi}w_ia_i)(\sum x_{hi}w_i)^{-1}$, $S = \sum_i \sum_h w_i x_{hi}(a_i - \bar{a}_h)^2$, and minimize S subject to $x_{hi} \geq 0, \sum_h x_{hi} = 1$. The solutions will always have just one of $x_{hi} = 1, h = 1, \ldots, G$, and so will solve the grouping problem. It is not clear whether or not this formulation simplifies the solution.

PROGRAMS

FISH partitions data, consistently with input order, to maximize between cluster sum of squares.

PFISH prints output from Fisher algorithm.

```
      SUBROUTINE FISH(X,SG,MG,N,K)
C.... X = M BY 1 VECTOR
C     PROGRAM GROUPS REAL VALUED OBSERVATIONS X(1),  . . X(N).
C     THE OBSERVATIONS NEED NOT BE ORDERED, BUT THE GROUPS WILL ALL CONSIST OF
C     SEQUENCES OF OBSERVATIONS X(I), . . X(J).
C     IF A PARTITION INTO K CLUSTERS IS REQUESTED, PARTITIONS LE K ARE AUTOMATIC
C     PRINT OUT INFORMATION ABOUT CLUSTERS BY CALLING PFISH
C.... X = N BY 1 ARRAY OF OBSERVATIONS TO BE FITTED
C.... MG = N BY K ARRAY,WHERE MG(I,J) IS LOWER BOUNDARY OF JTH GROUP, IN OPTIMAL
C          SPLIT OF X(1),... X(I) INTO J GROUPS.
C.... SG = N BY K ARRAY, SG(I,J) IS SUM OF SQUARES WITHIN GROUPS FOR
C          X(1),... X(I) SPLIT OPTIMALLY INTO J GROUPS.
C.....................................................................
      DIMENSION X(N),SG(N,K),MG(N,K)
C.... INITIALIZE SG,MG
      DO 20 J=1,K
      MG(1,J)=1
      SG(1,J)=0.
      DO 20 I=2,N
   20 SG(I,J)=10.**10
C.... COMPUTE SG,MG ITERATIVELY
      DO 30 I=2,N
      SS=0.
      S=0.
      DO 31 II=1,I
      III=I-II+1
      SS=SS+X(III)**2
      S=S+X(III)
      SN=II
      VAR=SS-S**2/SN
      IK=III-1
      IF (IK.EQ.0) GO TO 31
      DO 32 J=1,K
      IF (J.EQ.1) GO TO 32
      IF (SG(I,J).LT.VAR+SG(IK,J-1)) GO TO 32
      MG(I,J)=III
      SG(I,J)=VAR+SG(IK,J-1)
   32 CONTINUE
   31 CONTINUE
      SG(I,1)=VAR
   30 MG(I,1)=1
      RETURN
      END
```

```
      SUBROUTINE PFISH(X,SG,MG,N,K)
C.....20 MAY 1973
C.... USES OUTPUT FROM PROGRAM FISH TO PRINT CLUSTER DESCRIPTIONS
C.....
      DIMENSION SG(N,K),MG(N,K),X(N)
      WRITE(6,1) N,K
    1 FORMAT('1 PARTITION OF',I5,'  OBSERVATIONS UP TO ',I5,' CLUSTERS')
      DO 20 J=1,K
      JJ=K-J+1
      WRITE(6,2)JJ,SG(N,JJ)
    2 FORMAT('OTHE',I5,'  PARTITION WITH SUM OF SQUARES',F20.6)
      WRITE(6,3)
    3 FORMAT(' CLUSTER   NUMBER OBS   MEAN       S.D.   ')
      IL=N+1
      DO 21 L=1,JJ
      LL=JJ-L+1
      S=0.
      SS=0.
      IU=IL-1
      IL=MG(IU,LL)
      DO 22 II=IL,IU
      S=S+X(II)
   22 SS=SS+X(II)**2
      SN=IU-IL+1
      S=S/SN
      SS=SS/SN-S**2
      SS=(ABS(SS))**(0.5)
      WRITE(6,4) LL,SN,S,SS
    4 FORMAT(I5,5X,3F10.4)
   21 CONTINUE
   20 CONTINUE
      RETURN
      END
```