論文紹介

数理計画

M24　数理計画法における数値実験報告

H. P. Crowder, R. S. Dembo, 他. 316-329.

Mathematical Programming. **15**, 3, 1978.

この論文は，国際数理計画法学会に設けられたアルゴリズム委員会の手になるもので，(i)数理計画法におけるアルゴリズムのテストやコードの相互比較を行なう場合の基本的手続きはいかにあるべきか，(ii)(i)での研究結果の報告を主たる目的とする論文は，どの程度詳しい情報を読者に提供すべきか，(iii)専門誌のレフェリーが，これらの論文を受理もしくは棄却するにあたって判定の規準をどこに置くべきか，などについてきわめて詳細な提言を行なっている．これらの諸点は長い間良心的研究者を悩ませてきたものであり，何らかの規準の設定が待ち望まれていたが，著者らは，結果の再現可能性，統計的把握などの観点から多くの具体的な提案を行なっている．なお著者はもとより，この論文を掲載したMath. Prog. 誌の editor も，これはあくまでひとつの考え方に過ぎないことを断わっており，より多くの人々の意見を歓迎しているので，この分野に携わる者は内容について具体的に意見を述べるべきであると思われる．

(今野　浩)

確率統計応用

P25　異常値検出法のふるまいについて

P. Prescott. 10-25.

Appl. Statist. (*J. Roy. Statist. Soc. C.*) **27**, 1, 1978.

1組のデータについて，それに含まれている異常値(outlier)を検出する方法はいくつかあるが，異常値が存在する場合の検定統計量のふるまいは複雑で正確な分布を求めることは一般にむずかしく，従来シミュレーションによってしか調べられていない．著者はロバスト推定についての Tukey の感度曲線を2変数に拡張した感度曲面の等高線を描く方法で各検出法の特性の比較をしている．n 個のデータについての検定統計量Tの感度曲面は $T(a_1, a_2, \cdots, a_{n-2}, x_1, x_2)$で定義される x_1, x_2 の関数である．ここに$a_1, \cdots, a_{n-2}$はノーマル・スコアである．扱っている検出法は，データを $y_1<y_2<\cdots<y_n$ とするとき，(i)$y_n(y_1)$検出のための Grubbs 法：$D_n=S_n{}^2/S^2$ $(D_1{}^2=S_1{}^2/S^2)$，$\bar{y}=\sum_{i=1}^{n} y_i/n$，$S^2=\sum_{i=1}^{n}(y_i-\bar{y})^2$，$\bar{y}_n=\sum_{i=1}^{n-1} y_i/(n-1)$，$S_1{}^2=\sum_{i=2}^{n}(y_i-\bar{y}_1)^2$，$\bar{y}_1=\sum_{i=2}^{n} y_i/(n-1)$；(ii) y_1 または y_n 検出のための修正 Grubbs 法：$Dm_1=\min(S_1{}^2/S^2, S_n{}^2/S^2)$；(iii) y_n 検出のための Dixon 法：$r_{10}=(y_n-y_{n-1})/(y_n-y_1)$；(iv) y_1, y_n 検出のためのスチューデント化した範囲：$c_1{}'=(y_n-y_1/s$, $s^2=\sum_{i=1}^{n}(y_i-\bar{y})^2/(n-1)$；(v) y_n, y_{n-1} 検出のための Grubbs 法：$D_2=S_{n-1}{}^2{}_{,n}/S^2$, $S_{n-1}{}^2{}_{,n}=\sum_{i=1}^{n-2}(y_i-\bar{y}_{n-1,n})^2$, $\bar{y}_{n-1,n}=\sum_{i=1}^{n-2} y_i/(n-2)$；(vi) Rosner の逐次検定法などである．これらの等高線のグラフからあらかじめ定められた個数の異常値を検出するよう設計された方法((i)～(v))がどのような場合に masking effect を生じたり，1個の非常に大きな(小さな)異常値のために誤って多数の異常値があると判断させたりするかを説明し，これらの検出法の機械的使用の危険性を指摘するとともに逐次検定法の有効性を数値例も使って示している．

(柴田義貞)

コンピュータとシミュレーション

C 8　多数のイベントを必要とする離散系シミュレーションのためのイベントの取扱い方式

E. G. Ulrich. 777-785.

Comm. ACM **21**, 9, 1978

離散系のシミュレーションをコンピュータで行なうとして，実行すべきイベントがきわめて多数“予定表”に登録される可能性がある場合に，登録および検索を効率的に行なうための方法について論じている．“予定表”を表現するためのデータ構造として，従来から提案されている linear list 法と time-mapping 法とを結びつけて，converging lists 法と称するものを提案している．これは time-mapping 表(実行時刻と lists との対応表)と，数本の(横向きの) linear lists およびそれらが1点に集まってそこからさらに横に伸びる1本の linear list からなっている．各イベントは，実行時刻に応じていずれかの list に割当てられ，各 list 内では実行時刻の昇順に並ぶような位置に挿入される．単一の linear list を使う場合に比べて，この操作はずっと短時間で行なえる．同時刻に実行すべきイベントが2個以上登録されたときは，横向きの list から縦向きの list を伸ばし，これにつなぐ．

登録されたイベントを検索して実行する場合の問題についても論じている．その中には，同時刻に実行すべき複数のイベントの実行順序決定方法，登録時刻と実行時刻が同じイベントの処理，時間のスケールの動的変更等の問題が含まれている．

(伏見正則)