利失表を用いた決定理論の選択基準間の関係

吉川　歩[*1]・吉川　奈緒子[*2]

概要

意思決定問題の利失表を用いた代替案の選択方法として決定理論が利用されている．本稿では吉川により提案され拡張された 5 種の利失・後悔と 5 種の選択基準の 25 種の組合せについて，①得られる結果が同一となる冗長な条件と②適切に代替案の選択が行うことができない選択を阻害する条件を明らかにする．そしてそれらの冗長な条件と選択を阻害する条件を確認する方法を統計処理環境の R 言語を用いて実装する方法を示す．さらにそれらを用いた数値例により提案方法の有効性を示す．

キーワード：決定理論，利失表，選択基準，R 言語

1　まえがき

研究背景　経営戦略の選択などの意思決定では，将来の状況が複数存在し，かつどの状況になるかが選択時には不確定な中で，実際に代替案を実行する前に複数の代替案の中からいずれかを選択しなければならないという場面に遭遇することがある．当然ながらできるだけ利益が多い，あるいは損失が少ない代替案を選択できることが望ましい．この意思決定の方法の1つとして決定理論が用いられる．この決定理論では代替案を将来の状況の下で実行した場合の利益あるいは損失が何らかの方法で推定できると仮定する．この利益損失の推定値を代替案と将来の状況のマトリクスの形式で表したものを利失表あるいはペイオフマトリクス(*payoff matrix*)と呼ぶ．この利失表から代替案を決定するために選択基準が利用される[木下1996, 中村1997, 飯田2005]．この利失表と選択基準を用いた決定理論に関しては，経済産業省の国家資格である情報処理技術者試験のシラバスに含まれ，ほぼ毎年出題されている[JITEC]．また経営戦略の選択だけでなく，吉川らは低解像度ナンバープレート画像の数字の識別結果の判定法にも応用している[吉川ほか2012]．これらより，いわゆる枯れた手法ではあるが，利用範囲の広さと手法としての重要性が見て取れる．しかしながら，既に利用されている利失・後悔とそれに対する選択基準は利用可能なもののうちの一部である．この点に着目し，吉川は利失・後悔と選択基準の新たな組合せを提案している．[吉川2013a, 吉川2013b, 吉川2014a, 吉川2014b].

[*1] 甲南大学大学院社会科学研究科会計専門職専攻
[*2] 富士通株式会社

研究目的　吉川の提案により利用可能な利失と選択基準の候補は増えたが，利失・後悔と選択基準の組合せ相互の独立性に関する考察が不足している．つまり同じ利失表から出力される代替案が同一となる利失・後悔と選択基準の組合せは新たに付け加える意味が薄い．そこで本稿では，利用可能な利失・後悔と選択基準について相互の独立性を考察し，相互関係を明らかにすることを目的とする．まず2章では利失表の利失・後悔と選択基準の定義と記法を示す．続く3章では利失・後悔と選択基準の組合せの比較を行い，異なる組合せから得られる代替案の選択結果が一致する冗長な条件を明らかにする．また利失・後悔のマトリクスで代替案の選択が適切に行うことが困難な阻害条件も合わせて明らかにする．そして4章では統計解析環境であるR言語を用いて3章で示した条件の確認方法の実装とその処理の一例を紹介する．

2　利失表および選択基準

2.1　利失表の記法

利失表の記法　まず本稿で用いる利失表の記法について触れておく．m個の将来の不確定な状況($state\ of\ nature$)を$s_j(j=1,\ldots,m)$とする．それに対してn個の取りうる代替案($alternative$)を$a_i(i=1,\ldots,n)$とする．各状況s_jのもとで代替案a_iを行った時に得られる利失($payoff$)あるいは結果($consequence,\ outcome$)の推定値をc_{ij}とする．よってこの意思決定問題の利失表は表1のように表せる．なおp_jは各状況s_jの生起確率である．ただしこの生起確率p_jは必須ではない．また多くの場合，c_{ij}およびp_jは推定値であることに注意されたい．

表 1: 本稿で用いる利失表

状況		s_1	$\ldots$	s_j	$\ldots$	s_m
生起確率		p_1	$\ldots$	p_j	$\ldots$	p_m
代替案	a_1	c_{11}	$\ldots$	c_{1j}	$\ldots$	c_{1m}
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
	a_i	c_{i1}	$\ldots$	c_{ij}	$\ldots$	c_{im}
	$\vdots$	$\vdots$	$\ddots$	$\vdots$	$\ddots$	$\vdots$
	a_n	c_{n1}	$\ldots$	c_{nj}	$\ldots$	c_{nm}

後悔のマトリクス　表1の利失表を直接利用せずに，後悔($regret$)のマトリクスも利用される．代替案選択における後悔r_{ij}は“代替案選択後に状況が確定したとき，最良の代替案の結果と選択した代替案の結果の間の差異” つまり“ある状況s_jで最良の案を選べなかった損失”を表す．

$$r_{ij} = c_{ij} - \max_{i=1,\ldots,n} c_{ij} \tag{1}$$

なお文献によっては後悔を最大値からの差異により定義されている場合もあるが，本稿では(1)式を用いる．したがって後悔は必ず0以下となる．

状況内順位のマトリクス　代替案を決定する際には制御できない将来の状況が変化してもできるだけよい結果が得られる代替案を採択したい．このために従来は(1)式で定義される後悔のマトリクスr_{ij}が利用されていた．これに対し吉川は，いわゆる選び損ないではなく，想定した状況と異なったとしても良好な選択となることを積極的に表す量を状況内順位o_{ij}と呼び，(2)式で定義している[吉川2013b, 吉川2014a, 吉川2014b].

$$o_{ij} = \begin{cases} 1 & c_{ij} = \max\limits_{i=1,\ldots,n} c_{ij} \\ 0 & otherwise \end{cases} \tag{2}$$

(2)式は状況s_j内で各代替案a_iが一番よいかどうかを定量化していることになる．

レンジ補正後悔のマトリクス　また状況間で利失のレンジが大きく異なるときに後悔r_{ij}をそのまま利用した場合に，後悔のレンジが大きい状況のみで代替案選択が行われてしまう問題がある．そこで吉川は利失のレンジの影響を補正した後悔をレンジ補正後悔q_{ij}と呼び，(3)式で定義している[吉川2013b, 吉川2014a, 吉川2014b].

$$q_{ij} = \frac{r_{ij}}{\max\limits_{i=1,\ldots,n} c_{ij} - \min\limits_{i=1,\ldots,n} c_{ij}} \tag{3}$$

最大値補正後悔のマトリクス　状況間の利失の差異を補正する方法としては(3)式のレンジで補正を行うもの以外に最大値で補正する方法も考えられる．吉川は，利失の最大値を考慮した後悔を最大値補正後悔w_{ij}と呼び，(4)式で定義している[吉川2013b, 吉川2014a, 吉川2014b]．ただしw_{ij}を利用するためには状況ごとの最大値が必ず正でなければならい．

$$w_{ij} = \frac{r_{ij}}{\max\limits_{i=1,\ldots,n} c_{ij}}, \text{ただし} \max_{i=1,\ldots,n} c_{ij} > 0 \tag{4}$$

なお各利失・後悔を区別せずに一括して取り扱う場合には“$*_{ij}$”の標記で利失・後悔を表すものとする．

2.2　選択基準

選択基準の大別　選択基準を大別すると①生起確率あるいは確率の概念を利用するもの，②最小値，最大値に着目するものに大別することができる．なお従来の分類では③後悔を利用するものを別途区別していたが，利失・後悔を一括として扱うためにこの分類は行わない．①には，期待値基準，*Laplace*基準が，また②にはマクシマックス基準，マクシミン基準，*Hurwicz*基準がそれぞれ該当する．またマクシミン基準については，ミニマックス基準という表現も多くみられるが，代替案内の評価値の演算と代替案間の比較の演算で混乱を招く場合もあるため本稿ではマクシミン基準に統一する．なお吉川は従来の選択基準では代替案間の評価値の比較が最大値演算のみで行われている点に着目し，例えばミニミン基準のような最小値演算を用いる方法を試みている[吉川2013a].

数学的にはまったく無意味ではないが，あえて最小となる代替案を選ぶ合理的な根拠の欠如は否めないため，代替案間の比較演算は従来通り最大値演算を用いる．

期待値基準　表1の利失表で状況s_jの生起確率p_jが推測できるとき，各代替案の評価値e_iとして期待値を用い，最大となる代替案a_iを採択する方法である．(5)式で与えられる．

$$\max_{i=1,\ldots,n} \sum_{j=1}^{m} p_j \cdot *_{ij} \tag{5}$$

*Laplace*基準　表1で生起確率p_jは推測できないが，各状況s_jが等確率で発生するとみなし，期待値基準を適用する方法である．評価値として代替案ごとに平均値を算出している，あるいは利失の総和を評価値として用いているとみなすこともできる．(6)式で与えられる．

$$\max_{i=1,\ldots,n} \sum_{j=1}^{m} \frac{*_{ij}}{m} = \frac{1}{m}\left(\max_{i=1,\ldots,n} \sum_{j=1}^{m} *_{ij}\right) \tag{6}$$

マクシマックス基準　表1の生起確率は用いず，代替案ごとに状況間で最大の利失を求めて評価値e_iとし，さらに代替案間で最大となる代替案を採択する方法である．事実上，利失表中の最大利失を与える代替案を選択していることになる．このことより楽観的選択基準とも呼ばれる．(7)式で与えられる．

$$\max_{i=1,\ldots,n} \cdot \max_{j=1,\ldots,m} *_{ij} \tag{7}$$

マクシミン基準　表1の生起確率は用いず，代替案ごとに状況間で最小の利失を求めて評価値とし，さらに代替案間で最大となる代替案を採択する方法である．各代替案で得られる最小利失をもとにした選択であるため悲観的選択基準とも呼ばれる．また*Wald*基準と呼ばれることもある．(8)式で与えられる．

$$\max_{i=1,\ldots,n} \cdot \min_{j=1,\ldots,m} *_{ij} \tag{8}$$

*Hurwicz*基準　この基準では楽観度係数(*coefficient of optimism*):$\alpha(0 \leq \alpha \leq 1)$を導入し，代替案ごとの利失の最大値の$\alpha$倍と最小値の$(1-\alpha)$倍の和を求めて評価値とし，さらに代替案間で最大となる代替案を採択する方法である．αが1と0のときはそれぞれ前述のマクシマックス基準，マクシミン基準に一致する．(9)式で与えられる．

$$\max_{i=1,\ldots,n} \left\{\alpha \cdot \max_{j=1,\ldots,m} *_{ij} + (1-\alpha) \cdot \min_{j=1,\ldots,m} *_{ij}\right\} \tag{9}$$

3　選択基準間の冗長性と代替案の選択阻害条件

扱う問題点　2章で利失・後悔として5種，それらに対する選択基準として5種の演算を示した．これより25種の組合せが代替案の選択に利用可能である．ここで問題となるのは，これらの25種の手法から得られる選択結果の独立性あるいは従属性である．つまり利用可能な手法は増えたもの

の，得られる結果が複数の手法の間で同一であるのならば冗長性が増加したに過ぎない．まず得られる結果が一致する組合せを明らかにすることが必要となる．またそれに加えて利失・後悔のマトリクスが特定の性質を有するとき代替案の選択が適切に行えない場合がある．先の問題に加えて代替案の選択を阻害する利失表の条件についても明らかにすることも重要である．よって扱う問題をまとめると次の2点となる．

Ⅰ 代替案の選択結果が一致する利失・後悔と選択基準の冗長な組合せの条件

Ⅱ 代替案の適切な選択を阻害する利失・後悔の条件

3.1 利失・後悔と選択基準間の冗長性

冗長性の考え方　利失表と(5)から(9)式までの選択基準を組合せた代替案の選択での結果は代替案a_iの評価値e_iの最大値として与えられる．そのため評価値e_iの最大値を与える代替案a_iが一致する組合せは冗長であるという考え方もできる．本稿ではこれを広義の冗長性と呼ぶ．その一方で，単に最良案だけでなく，代替案の評価値e_iの値や順序の情報も得たいときもある．この場合は複数の選択基準から得られる評価値e_iの順序あるいは値が一致した場合が冗長であるとみなされる．本稿ではこれを狭義の冗長性と呼ぶ．以下ではまず後者の狭義の冗長性について考察し，次いで広義の冗長性に緩和して考察を行う．

2種の冗長性　選択基準から計算される代替案ごとの評価値e_iの順序あるいは値が一致してしまうケースには，解析的に一致する場合と数値的に一致する場合に大別できる．前者はさらに条件によらず一致する場合と一定の条件を満たす場合に一致する場合に細分できる．以下ではそれぞれについていずれの組合せで生じるのかを示していく．

条件によらず解析的に一致　この分類に該当する組合せは常に冗長であり，いずれか一方の評価を行えばよい．次のような組み合わせが該当する．

① 利失c_{ij}と後悔r_{ij}の期待値基準および$Laplace$基準

② 後悔r_{ij}，状況内順位o_{ij}，レンジ補正後悔q_{ij}，最大値補正後悔w_{ij}に対するマクシマックス基準

利失c_{ij}と後悔r_{ij}の期待値基準および*Laplace*基準　この条件は中村[中村1997]が指摘しているものである．これは次のように示すことができる[吉川2014a]．ある代替案a_iに対する後悔期待値基準の評価値e_iは(5)式から(10)式の第2式となる．

$$e_i = \sum_{j=1}^{m} p_j \cdot r_{ij} = \sum_{j=1}^{m} p_j \cdot c_{ij} - \sum_{j=1}^{m} p_j \cdot \max_{i=1,\ldots,n} c_{ij} \tag{10}$$

後悔r_{ij}の定義(1)式から利失c_{ij}で書き直すと第3式のようになる．ここで(10)式の第3式の第2項はiによらず一意であるため，後悔の期待値基準は利失の期待値基準にバイアス項が加わったにすぎない．当然，$Laplace$基準も期待値基準の特殊事例であるため，同様の結果となる．なお順位，値で

一致するため狭義の冗長性となる．

後悔r_{ij}，状況内順位o_{ij}，レンジ補正後悔q_{ij}，最大値補正後悔w_{ij}に対するマクシマックス基準利失c_{ij}を除く他の利失および後悔は定義に状況s_jごとの利失c_{ij}の最大値$\max_{i=1,\ldots,n} c_{ij}$から偏差を利用している．そのためマクシマックス基準を適用する際の評価値e_iの演算

$$e_i = \max_{j=1,\ldots,m} *_{ij}$$

を行った結果の最大値を与える代替案a_iが一致することになる．なお最大値以外の評価値e_iの順位は利失および後悔により異なる場合があるため狭義の冗長性は満たされない．

一定の条件を満たす場合に解析的に一致　この分類に該当する組合せは利失・後悔が一定の条件となった場合に，いずれか一方の評価を行えばよい．該当するものとしては次のようなものがある[*3]．

③ 利失c_{ij}の状況s_jごとの最大値$\max_{i=1,\ldots,n} c_{ij}$がすべて等しい場合，利失$c_{ij}$，後悔$r_{ij}$，最大値補正後悔$w_{ij}$に同じ選択基準を用いた評価値の順位が一致

⑤ 利失c_{ij}の状況s_jごとの最大値と最小値の差$\max_{i=1,\ldots,n} c_{ij} - \min_{i=1,\ldots,n} c_{ij}$がすべて等しい場合，後悔$r_{ij}$とレンジ補正後悔$q_{ij}$に同じ選択基準を用いた評価値の順位が一致

⑥ 利失c_{ij}の状況s_jごとの最大値がすべて等しくかつ最小値もすべて等しい場合，すべての利失および後悔に状況内順位o_{ij}を除いた同じ選択基準を用いた評価値の順位が一致

上記の③から⑥の各条件を考えるときに利失c_{ij}の最大値，最小値および両者の差をそれぞれ

$$\max_{i=1,\ldots,n} c_{ij} = \beta_j, \qquad \min_{i=1,\ldots,n} c_{ij} = \gamma_j, \qquad \beta_j - \gamma_j = \delta_j$$

と表す．そうすると後悔r_{ij}，レンジ補正後悔q_{ij}，最大値補正後悔w_{ij}は(1)式，(3)式，(4)式より，それぞれ

$$r_{ij} = c_{ij} - \beta_j, \qquad q_{ij} = \frac{r_{ij}}{\beta_j - \gamma_j} = \frac{r_{ij}}{\delta_j}, \qquad w_{ij} = \frac{r_{ij}}{\beta_j}$$

と表すことができる．またそれぞれが状況s_jによらず一定，つまりjによらず一定となることを定数K_β，K_γ，K_δを使って

$$\beta_j = K_\beta, \qquad \gamma_j = K_\gamma, \qquad \delta_j = K_\delta$$

のように表すと，例えば③の条件は

$$r_{ij} = c_{ij} - K_\beta, \qquad w_{ij} = \frac{r_{ij}}{K_\beta}$$

と書き換えることができる．これより利失c_{ij}に定数を加えたもの，さらにそれを定数で割ったものになっているため，同じ選択基準を適用した場合の評価値の順位が一致することがわかる．残りの⑤から⑥の条件も同様にして導くことができる．

[*3] なお条件④利失 c_{ij} の状況 s_j ごとの最小値 $\min_{i=1,\ldots,n} c_{ij}$ がすべて等しい場合，レンジ補正後悔 q_{ij} と最大値補正後悔 w_{ij} に同じ選択基準を用いた評価値の順位が一致は成立しない場合が後日判明したため削除した

数値的に一致する場合　前述の解析的に一致する場合は定義式から複数の組合せで評価値の順位が一致する条件を導くことができた．しかしながらそれらの条件を満たさない場合でも数値的に複数の組合せで評価値の順位が一致する場合がある．異なる利失・後悔に同じ選択基準を適用した結果が一致するケースを予め検出することで無駄な評価を回避することができる．この検出法として次の方法が利用できる．

⑦ 利失・後悔のマトリクスを1次元化して順位相関係数を算出

⑦の条件は，先に示した③から⑥ の条件を一般化したものとなっている．なお相関係数の算出に順位相関係数を用いるのは，異なる利失・後悔の間に線形関係がある必要はなく，単調性があれば十分であることによる．したがって$kendall$あるいは$spearman$の順位相関係数が1となれば，いずれか一方の利失・後悔のみを利用すればよい．

3.2　代替案の選択を阻害する条件

代替案の選択を阻害する利失あるいは後悔の条件　3.1では利失・後悔のマトリクスが特定の条件を満たす場合に同じ代替案の選択が行われる問題点について示した．それとは別に，利失・後悔が特定の特徴を有する場合には代替案が選択できなくなる場合がある．これには次のような条件が該当する．

1. 代替案a_iごとの利失・後悔の最大値$\max_{j=1,\ldots,m} *_{ij}$がすべて等しいとき，マクシマックス基準で代替案を選択不能
2. 代替案a_iごとの利失・後悔の最小値$\min_{j=1,\ldots,m} *_{ij}$がすべて等しいとき，マクシミン基準で代替案を選択不能
3. 代替案a_iごとの利失・後悔の最大値がすべて等しく，かつ最小値もすべて等しいとき，$Hurwicz$基準で代替案を選択不能
4. 代替案a_iごとの利失・後悔の総和$\sum_{j=1}^{m} *_{ij}$がすべて等しいとき，$Laplace$基準で代替案を選択不能
5. 各代替案a_iの利失・後悔に状況s_j内の最小値$\min_{i=1,,\ldots,n} *_{ij}$が少なくとも1つ存在するとき，レンジ補正後悔$q_{ij}$ではマクシミン基準で代替案を選択不能
6. 状況内順位o_{ij}ではマクシミン基準で代替案を選択不能

1から4の各条件はそれぞれの選択基準の定義の(6)式から(9)式より直ちに導かれる．5の条件はレンジ補正後悔q_{ij}は定義より必ず$-1 \leq q_{ij} \leq 0$となるため，最小値が少なくとも1つ存在すればマクシミン基準の各代替案ごとの評価値は-1となってしまうためである．また6の条件は状況内順位o_{ij}の定義より最大値以外の利失に対しては0となるためである．マクシミン基準での評価値がすべて0でなければどの状況でも最大値となる代替案が存在することになる．

冗長条件および阻害条件のまとめ　以上の結果をまとめると表2のようになる．

表 2: 代替案選択の冗長性と阻害条件：表内の数字は前掲の各条件を表す

	c_{ij}	o_{ij}	r_{ij}	q_{ij}	w_{ij}	$*_{ij}$
マクシマックス基準		②	②	②	②	$\boxed{1}$
マクシミン基準		$\boxed{6}$		$\boxed{5}$		$\boxed{2}$
Hurwicz 基準						$\boxed{3}$
期待値基準	①		①			
Laplace 基準	①		①			$\boxed{4}$
選択基準共通	③⑥		③⑤⑥	⑤⑥	③⑥	⑦

$\boxed{4}$ R 言語による冗長性と選択阻害性の確認

4.1 R 言語を用いた冗長性と選択阻害性の確認

R言語を用いた実装　吉川は利失c_{ij}とその生起確率p_jから代替案を選択する手順を統計処理環境のR言語を用いて実装している[吉川2014a]．詳細は付録Aに掲載してあるのでそちらを参照されたい．本稿でも吉川の定義した関数を利用し，R言語で3章で示した冗長性と選択阻害性を確認する演算を定義する．なおR言語の関数等については紙幅の都合上割愛する．関連資料を参照されたい[R-project]．

冗長性の確認　3.1で示した冗長条件のうち，①および②は定義式から常に成立するため冗長となる組合せ内でいずれか1つを用いればよい．したがってここでは扱わず，残りの③から⑦の条件のみを扱う．③から⑥の条件はいずれも状況s_jごとの利失c_{ij}の最大値β_j，最小値γ_j，両者の差δ_jがすべて一致するかをチェックすればよい．そのためには次のような要素を一致する関数“$Equiv()$”を定義し，引数として渡す配列を条件に合うように前処理すればよい．

- 要素の一致（すべて一致した時は“$TRUE$”，そうでなければ“$FALSE$”を戻す）

```
Equiv = function(mt, k=mt[1]){
  for(i in 1:length(mt)){
    if(k != mt[i]) return(FALSE)
  }
  return(TRUE)
}
```

  第2引数は省略可能であり，省略時は第1引数の最初の要素を自動的に利用

- 条件③用：“c”にセットした配列の列ごとの最大値がすべて等しいとき戻り値“$TRUE$”

```
Equiv(apply(c, 2, max))
```

- 条件⑤用：“c”にセットした配列の列ごとの最大値と最小値の差がすべて等しいとき戻り値“$TRUE$”

```
Equiv(apply(c, 2, max) - apply(c, 2, min))
```

- 条件⑥用："c"にセットした配列の列ごとの最大値がすべて等しく，かつ最小値もすべて等しいとき戻り値"$TRUE$"

```
Equiv(apply(c, 2, max)) && Equiv(apply(c, 2, min))
```

また⑦の条件については次のようにチェックできる．

- 条件⑦用："$c1, c2$"にセットした利失・後悔が冗長なときベクトルに変換して求めた順位相関係数が1

```
cor(as.vector(c1), as.vector(c2), method="kendall")
```

なお$method$ ="$spearman$"でも可．

選択阻害性の確認　他方3.2で示した代替案の選択が阻害される条件のうち，$\boxed{6}$は定義から導かれるためチェックの必要はない．他の$\boxed{1}$から$\boxed{4}$の条件は付録Aに示した吉川の選択基準の関数と前述の要素の一致をチェックする関数"$Equiv()$"を利用すれば簡単に確認できる．

- 条件$\boxed{1}$用："c"にセットした配列の行ごとの最大値がすべて等しいとき戻り値"$TRUE$"

```
Equiv(Max(c))
```

- 条件$\boxed{2}$用："c"にセットした配列の行ごとの最小値がすべて等しいとき戻り値"$TRUE$"

```
Equiv(Min(c))
```

- 条件$\boxed{3}$用："c"にセットした配列の行ごとの最大値がすべて等しく，かつ最小値がすべて等しいとき戻り値"$TRUE$"

```
Equiv(Max(c)) && Equiv(Min(c))
```

- 条件$\boxed{4}$用："c"にセットした配列の行ごとの総和がすべて等しいとき戻り値"$TRUE$"

```
Equiv(Laplace(c))
```

条件$\boxed{5}$に忠実に関数を作成することも可能であるが，定義済みの関数を流用する方が容易である．

- 条件$\boxed{5}$用：利失から求めた状況内順位をセットした配列の行ごとの最小値がすべて等しいとき戻り値"$TRUE$"

```
Equiv(Min(OrderValue(c)))
```

これらからわかることは，選択基準により評価値e_iを求めたときに，上記の条件$\boxed{1}$から$\boxed{5}$に該当しない場合でも，常に評価値の一致チェックを関数"$Equiv()$"を使って行っておけばよいということである．

4.2 R 言語による演算例

R言語による演算例　4.1で定義した関数を用いて演算を行った一例を次に示す．まず冗長性のチェックから示す．以下の例では，利失を$c1$という3行×3列の行列に代入し，それぞれ行名と列名を設定している[*4]．

```
> ncol=3; nrow=3; byrow=T↵
> dimnames=list(paste("a",1:nrow,sep=""), paste("s",1:ncol,sep=""))↵
> c1=matrix(c(40,0,40,0,40,30,10,20,10), nrow, ncol, byrow, dimnames)↵
> c1↵
   s1 s2 s3
a1 40  0 40
a2  0 40 30
a3 10 20 10
```

まず$c1$に冗長性のチェックの条件①を適用している（冗長となるような例を用いている）．

```
> Equiv(apply(c1,2,max))↵
[1] TRUE
```

異なる利失・後悔間で同じ選択基準を用いて評価値を求めたときに順位が一致するはずである．ここでは後悔$r1$を求めている．

```
> r1=Regrets(c1)↵
> r1↵
    s1  s2  s3
a1   0 -40   0
a2 -40   0 -10
a3 -30 -20 -30
```

試しに両者からマックス演算で評価値を求め，順位が一致することを相関係数で確認している．ミニ演算でも同様の結果が得られている．

```
> Max(c1)↵
a1 a2 a3
40 40 20
> Max(r1)↵
 a1  a2  a3
  0   0 -20
> cor(as.vector(Max(c1)),as.vector(Max(r1)))↵
[1] 1
> cor(as.vector(Min(c1)),as.vector(Min(r1)))↵
[1] 1
```

[*4] なお “↵” はキーボードから Enter キーを入力していることを表す．また行頭の “>” はプロンプトであり入力する必要はない．

次に選択阻害性の例について示す．同様に$c2$に利失を入力する．

```
> c2=matrix(c(40,10,30,10,40,30,40,30,10),nrow,ncol,byrow,dimnames)↵
> c2↵
   s1 s2 s3
a1 40 10 30
a2 10 40 30
a3 40 30 10
```

この例の場合，選択阻害条件の$\boxed{3}$に該当するように設定しているので，楽観度係数$\alpha = 0.5$で$Hurwicz$基準を適用した場合に代替案が選択できなくなっていることがわかる．

```
> Equiv(Max(c2)) && Equiv(Min(c2))↵
[1] TRUE
> Hurwicz(c2,0.5)↵
a1 a2 a3
25 25 25
```

5 まとめ

得られた結果　吉川[吉川2013b, 吉川2014a, 吉川2014b]により拡張された利失・後悔と選択基準について，利失・後悔と選択基準の組合せが冗長となる条件を示した．また代替案の選択が阻害されてしまう利失・後悔の条件についても明らかにした．そして両者の条件をチェックする方法を統計処理環境のR言語を用いて実装した例と実行例についても紹介した．

参考文献

[飯田2005]　飯田耕司: 意思決定分析の理論, pp.153–158, 三恵社, 2005

[JITEC]　情報処理技術者試験センター: http://www.jitec.jp/

[木下1996]　木下栄蔵: わかりやすい意思決定論入門, pp.5–9, 近代科学社, 1996

[中村1997]　中村雅章: 経営意思決定手法の基礎, pp.147–155, 同友館, 1997

[R-project]　Rプロジェクトウェブページ: http://www.r-project.org/index.html

[吉川ほか2012]　吉川歩ほか: 低解像度ナンバープレート数字識別規則の改良–第7報–, 日本法科学技術学会誌, Vol.17(Supplement), pp.101, 2012

[吉川2013a]　吉川歩: 利失表を用いた決定理論における選択基準の比較, 第50回日本経営システム学会全国研究発表大会講演論文集, pp.32–35, 2013

[吉川2013b]　吉川歩: 利失表を用いた決定理論の選択基準に関する考察, 第29回ファジィシステムシンポジウム講演論文集, pp.153–156, 2013

[吉川2014a]　吉川歩: 利失表を用いた決定理論に関する一考察, 甲南会計研究, Vol.8, pp.87–100, 2014

[吉川2014b]　吉川歩: 利失表を用いた決定理論に関する一考察, 第52回日本経営システム学会全国研究発表大会講演論文集, pp.36–39, 2014

付録 A　R による利失，後悔，選択基準の実装

利失・後悔の計算　吉川[吉川2013b]が示しているRを用いて利失，後悔，選択基準を実装した例を参考まで掲載する．ここでは利失表の利失c_{ij}と生起確率p_jのみを利用者が入力するものとする．まず最初に後悔r_{ij}，状況内順位o_{ij}，レンジ補正後悔q_{ij}，最大値補正後悔量w_{ij}を求める関数を定義する．利失の行列mtを引数として各量を求める関数は以下のように定義できる．戻り値は行列となる．

- 後悔

```
Regrets = function(mt) sweep(mt, 2, apply(mt, 2, max))
```

- 状況内順位

```
OrderValue = function(mt) (Regrets(mt) >= 0)*1
```

- レンジ補正後悔

```
RangeModRegrets = function(mt){
  sweep(Regrets(mt), 2, apply(mt, 2, max) - apply(mt, 2, min), FUN="/")
}
```

- 最大値補正後悔

```
MaxModRegrets = function(mt){
  if(min(apply(mt, 2, max)) > 0)
    sweep(Regrets(mt), 2, apply(mt, 2, max), FUN="/")
}
```

代替案ごとの評価値の計算　選択基準を適用する際には，上記で計算された利失・後悔の行列を用いて代替案a_iごとに評価値を計算し，それらを代替案間で比較して最大値を与える代替案を決定する．利失あるいは後悔の行列をmt，生起確率をp，楽観度係数をaとすると以下のように定義できる．戻り値は名前付き数値ベクトル形式となる．

- 期待値

```
ExpectVal = function(mt, p){
  apply(sweep(mt, 2, p, FUN="*"), 1, sum)
}
```

- *Laplace*

```
Laplace = function(mt) apply(mt, 1, mean)
```

- マックス

```
Max = function(mt) apply(mt, 1, max)
```

- ミン

```
Min = function(mt) apply(mt, 1, min)
```

- *Hurwicz*[*5]

```
Hurwicz = function(mt, a){
  if(length(a) > 1){
    temp = matrix(0, ncol=length(a), nrow=dim(mt)[1])
    rownames(temp) = rownames(mt)
    colnames(temp) = paste(a)
    for(i in 1:length(a)){
      temp[,i] = a[i] * Max(mt) + (1-a[i]) * Min(mt)
    }
  }
  else temp = a * Max(mt) + (1-a) * Min(mt)
  return(temp)
}
```

代替案間の最大値の計算　代替案ごとの評価値の計算と代替案間の最大値の計算を分離しているため，代替案間の最大値を求める計算は共通して利用できる．代替案間の最大値の計算は次のように定義できる．

- マクシ

```
Maxi = function(mt) mt[which(mt == max(mt))]
```

元々の利失・後悔の行列の行名(rownames)に代替案名が定義されている場合は代替案名を“*names*”属性とする名前付き数値が，定義されていない場合は値のみが戻り値となる．最大値を取る代替案が複数となった場合はベクトルとして返される．

[*5] なお *Hurwicz* に関しては楽観度係数 a が単独の数値であっても，数値列（ベクトル）であっても対応可能なようにしている．