【原　著】

試験問題統計情報のデータベース化と利用環境の整備

大津　起夫*
石岡　恒憲*
橋本　貴充*

要　約†

昭和 54 年から平成 20 年度までの共通第 1 次学力試験および大学入試センター試験の本試験について，試験問題の検討に重要な統計量を求めた．本試験の各大問および各設問について，得点率，科目得点との相関などを求め，Web ブラウザにてこれらの統計情報および設問解答率分析図を閲覧可能とした．また，該当する試験問題文をもオンラインで閲覧可能とした．さらに，研究開発部ローカルエリアネットワーク内にセンター試験問題の全文検索システムを設置し，検索結果に関係する統計情報を参照可能とした．

整備されたデータベースに収録された過去の試験における統計量を検討すると，プレテストを行わずに問題の難易度を調整することが，かなり難しいことが伺える．

キーワード：データベース，難易度，識別力，テキスト検索

1　はじめに

大学入試センターによる全国統一の大学入学試験は，昭和 54(1979) 年に共通第 1 次学力試験が開始して以来，平成 21(2009) 年で 31 回目の実施を迎える．平成 2(1990) 年度以降は制度変更され大学入試センター試験となり，科目選択の自由度が増えた．大学入試センター試験となって以降は，アラカルト方式の科目選択が可能であるため，受験者数は科目により異なるが，もっとも受験者数の多い教科である外国語（英語を含む 5 科目）については平成 19(2007) 年では約 50 万 5 千人，平成 20(2008) 年には約 49 万 7 千人の受験者があった．

これまでに毎年度実施されている試験において出題された設問への受験者の解答データは，個人情報を秘匿した研究用のファイルとして大学入試センター研究開発部で研究に利用されてきた．しかしながら，データの量が膨大であること，また従来成績データの業務処理が古典的な大型計算機システムによって行われてきたこともあり，これらのデータは特別な電子的アノーテーションのないテキストファイルとして保存されており，ファイル中に含まれる情報を取り出すには，紙媒体のファイルフォーマット資料を参照する必要がある．このため特定の設問項目の正答率などの統計情報を導出するのはかなり煩雑な作業となる．本稿は，これらの成績データから統計情報を抽出し，試験問題の作成のため容易に参照を可能とした成果を報告するものである．

平成 9 年以降の大学入試センター試験の解答データについての統計情報の整備は，研究開発部にお

* 大学入試センター 研究開発部 試験評価解析研究部門
2008 年 11 月 28 日　受理
† 本研究は大学入試センター研究開発部におけるプロジェクト研究 IV 「センター試験に関わる統計情報の蓄積と利用に関する研究」（平成 18 年度～平成 20 年度）の成果に基づく．

ける「試験問題統計情報の整備に関する研究」(平成 14 年度～16 年度共同研究 I) (石塚他, 2005) 等のプロジェクトにおいて既に行われている．この研究では，平成 9 年から 16 年までの各科目の本試験について設問別の正答率，無解答率，設問解答率分析図（科目得点群別の選択肢選択比率の図），2 パラメータ項目反応モデルによる推定パラメータなどを導出し，さらに結果をデスクトップデータベースに搭載し，各種の統計グラフの閲覧と簡単な検索を可能とした．

本研究は，これまでの研究成果を受け継ぎ，さらに情報整備の範囲を拡大するものであるが，以下の様にいくつかの点で上記のプロジェクトとは異なる方針をとった．

1. 統計情報整備の目的の一つは，教育学的および心理学的な研究資料としての利用であるが，実用上最も重要な目的は試験問題作成時の参考資料としての利用である．前回のプロジェクトによって整備された資料は，大変に詳細であり，また多くの統計グラフを伴うかなり複雑な仕様のデータベースとして実現されていた．しかし，試験問題作成時の利用を想定すると，より単純な構成で掲載情報をより少数の重要な指標に絞ることが実用的であると判断された．このため整備する統計情報を精選し，情報の単純な呈示方法を考案した．
2. 最近の大学入試センター試験問題の採点においては，部分点の付加はほとんど行われていないが，平成 8 年 (1996 年) 以前の資料を検討すると部分点の付与が行われている場合がしばしば生じている．前回のプロジェクトにおいて作成された集計システムにおいては，部分点の付与についてあまり考慮していなかったが，多様な部分点付加方法に対応するよう仕様を変更した．

   統計情報の導出という目的に限定するならば，ごく少数の部分点の付与を正確に再現することにはさして重要性はないが，配点規則の復元誤りを検出するためには，すべてのケースにおいて正しく採点が行われることが望ましい．このため，部分点の付与をすべて再現する労力を払った．
3. 作成された情報の利用を考慮すると，個別の PC にシステムを配置することは管理上煩雑であり，ローカルネットワークを介し，サーバコンピュータ上の資料を Web ブラウザを用いて閲覧することが望ましい．利用者 PC 上にそれぞれデータをインストールして利用を行うことは不可能ではないが，データの更新を行うことが煩雑になり，運用上の困難をもたらす．このため統計情報はサーバで一括管理し，リモート PC から閲覧を可能にした．
4. これまでに統計情報から該当する試験問題文を参照することが可能であったが，これに加え試験問題文の全文検索機能との統合を行い，試験問題文の検索から関係する統計情報の参照を可能にした．

これら 4 点の実現を目的として，試験問題統計情報を導出し，またその閲覧システムを構築することとした．なお，ここで報告する統計情報は，昭和 54 年から平成 20 年度までの共通第 1 次学力試験および大学入試センター試験の本試験のうち，活字問題についてのものである．一部，点字での出題も行われているが，一部設問内容が異なるものがあるためここでの集計からは除いた．

以下ではプロジェクトの各側面について概要を説明する．前半では，センター試験成績ファイルからの統計量の導出の解説を行う．特に採点パラメータの同定方法と，プログラムにおける採点処理方法について説明する．いささか煩雑ではあるが，同種の採点プログラムを作成する場合の例として参照されることを考慮し，技術的な詳細をも含め説明する．

後半では，導出された統計情報を用いて閲覧用の XHTML ファイルを作成するための方法，また，Web サービスを利用してこれらを閲覧し統計情報を図示する方法，および試験問題文の全文検索システムの利用方法の概要について説明し，加えて過去資料における統計情報の特徴について紹介する．

2 採点規則の推定

大学入試センターが実施する試験は，問題と正解およびその配点情報が実施後に公開されるため，これらの情報を用いれば原則として採点方法を復元することが可能である．ただし，これらの試験のうち平成 8 年以前の正解表においては，配点情報が設問単位で示されている．このため設問が複

数の解答選択肢の記述欄から構成されている場合には，詳細な配点情報は正解表からは明らかでないことがある．また，過去においては，正解表に示されていないが，一部の問において部分的に得点を与えている場合もあるため，公開された正解表の情報のみでは，完全には採点規則を復元することができない．

このため，科目別得点を計算するための処理において生成された作業用の電子ファイル（「成績ファイル」）を利用して，これらの詳細な情報を推定することにした．

成績ファイルは，マークシートへの解答とそれらの項目別の採点結果，およびその合計値である科目別得点が記述されたファイルである．この電子ファイルは採点処理のための業務の 1 段階で得られるものであるが，大学入試センター研究開発部には，個人情報を削除し匿名化した成績ファイルが研究用に保存されており，これを用いることにより，マークシートへの解答とそれに対応する項目別の得点の関係を推定することができる．

個別の配点規則の同定は次のような手順で行った．

1. 成績ファイルから，各科目について本試験で活字問題を受験したもののレコードを抽出する．
2. 成績ファイル中に記録されたマークシートの各解答欄について選択肢（数字指定の場合も含め）の選択頻度を求める．
3. 成績ファイル中には各解答欄に対応する項目別得点欄があるが，それらの頻度分布を求める．
4. 前項で得られた項目別得点頻度と，公表された正解表とを比較し，発表済み正解のほかに部分点が与えられているかを確認する．
5. さらに部分点の配点がある場合には，次のような手順で部分点の採点規則を推測する．
   (a) 採点が 1 カラムを対象としている場合には，前項の得点頻度と，2 で得られたマークシートの選択肢の頻度とを比較し，部分点の付与された選択肢を推定する．
   (b) 組み合わせ採点（複数の解答欄への解答の組み合わせによって配点が定まる場合）では，該当するマークシート解答と項目別得点との同時頻度を求め，配点規則を推測する．
6. 上記の手続きによって推定された配点規則を用いて採点を行い，合計点が科目別得点と一致することを確認する．
7. もし，科目別得点と一致しないレコードが存在する場合には，それらに共通する性質を検討することにより修正箇所を確認し，配点規則を修正した後に再び採点を行い，科目別得点との一致を確認する．

以上の手続きによって採点規則を推定したところ，平成 2(1990) 年から平成 19(2008) 年までの大学入試センター試験（本試験活字問題）については，12 件を除いて科目別得点と一致することが確認できた．これらの不一致の生じた答案においては，特定の解答欄がダブルマークされているにも関わらず得点が与えられている（通常ではダブルマークはゼロ点となる）．またこの設問には正解のほかに別解があり，複数の解に配点が与えられる．これらを検討すると，ダブルマーク解答については，マークシートへの記述にさかのぼり，記入箇所を確認して得点を与えたものと推測される．しかし，本データベースの収録においては，これらの項目には得点を与えず，不正解とみなして統計処理を行った．また，共通第 1 次学力試験については，残念ながら現在のところ部分点の配点を完全に再現することができない部分があり，約 500 件ほどの不一致を残している．配点食い違いの最大は 4 点である．

上記の手続きによって得られた採点規則は，データから推測されたものであり，指示された採点規則と若干異なる可能性もあるが，受験者が大量である場合には，ほぼこれらの結果は一致すると思われる．ただし，受験者が少数である場合には，正解表に示された以外に別解あるいは部分点を与える指示があったとしても，実際の解答中に該当するものが存在しないと，これらの規則を推測することができない．

これらの作業を効率的に行うために，カラムごとの頻度一覧を表示するための小規模なプログラムを C++によって記述した．これらの集計操作は汎用の統計ソフトウェアを用いても可能であるが，データが大量であること，および作業が頻繁に発生するため，これらに特化した小規模なプログラムを用いるのが効率的であった．

3 採点プログラムの構成

共通第1次学力試験とセンター試験を併せて平成20年度までに30回の試験が実施されているが，過去分の採点方式の中には，かなり複雑な組み合わせ配点規則を要求するものがある．特に複雑な規則は，共通第1次学力試験での部分点の配点に多い．平成2年度以降の大学入試センター試験においてはより簡明な形式に採点方式が整理されているが，なお数種類の組み合わせ配点方式が用いられており，単純な1カラムの記入文字と正解との対応関係によっては採点できない場合が存在する．このプロジェクトにおいては，共通第1次学力試験も含め，試験問題統計情報を統合する必要があるため，大学入試センターによる過去の試験に現れた採点方式を全て再現する必要がある．

これに対応するために，次のような採点プログラムの設計方針をとった．ただし，ここで示す方法は，過去の採点を再現するため研究として行うものであり，実際にセンター試験業務で行われている採点処理とは異なる．

1. 採点処理は,「基本アイテム」,「採点単位」,「大問」,「冊子」の4つの階層から構成する．
2.「基本アイテム」は，特定のカラム（1つあるいは原則として連続した同幅の複数のカラム）に記述された解答を読み取り，対応する得点を定義する．「基本アイテム」は，採点プログラムにおいてはC++言語で記述されたクラスオブジェクト（クラス名 `dncItem`）であり，指定されたカラムの解答を文字列のベクトルとして入力し，採点処理を行う．また複雑な部分点の配点を実現するために，負の配点も許す．
3.「採点単位」は，1つ以上の「基本アイテム」から構成される．通常は1つの「基本アイテム」からなり，その得点が「採点単位」の得点となる．ただし，複雑な組み合わせ採点の場合（組み合わせ採点によって与えられる得点が，個別のカラムへの得点和ではない場合など）には，複数の「基本アイテム」によって一つの「採点単位」が構成される．この場合，これらの「基本アイテム」の得点和または零点のいずれか大きい方によって「採点単位」の得点を定義する．このため「基本アイテム」の得点は負となりうるが,「採点単位」の得点は常に非負となる．

   「基本アイテム」にはC++のクラスが対応しているが，現状では「採点単位」に直接対応するC++の固有のクラスは存在せず,「基本アイテム」へのポインタのベクトル（`vector< dncItem* >`型）として表現する．

   後述するように，今回収録した範囲の試験問題の多くについては,「採点単位」の構造を用いずとも，一つの（組み合わせ得点を含む）設問項目に対して，一つの「基本アイテム」を対応させることにより，採点を実現することは可能である．しかしながら，前回のプロジェクトにおいて蓄積されたパラメータ書式を利用して「基本アイテム」の属性を定義するさいに困難な場合があり，また例外的に複雑な組み合わせ採点も存在するため，複数の「基本アイテム」から構成される「採点単位」を設定して対応を容易にした．
4.「大問」は，各科目の試験における大問（第1問，第2問など）を表現するものであり,「採点単位」の集合が対応する．「大問」も「採点単位」と同じく，C++のクラスに直接には対応しない．現状ではC++プログラム中で,「基本アイテム」のポインタベクトルとして表現している．
5.「冊子」は，1科目の試験全体を表す．これにはC++のクラス（`dncBooklet`）のオブジェクトとして表現される．このオブジェクトは，一つの試験科目の問題冊子を構成する「基本アイテム」のベクトルをメンバー変数として保持する．また「採点単位」および「大問」の構造についての情報を，項目番号あるいは大問番号をキーとし,「基本アイテム」のポインタのベクトルを値とするマップコンテナによって表現する．これらの対応関係を表すには，項目番号および大問番号が整数であるため，ベクトルコンテナ，あるいは配列によって表現することも可能ではあるが，これらの識別記号が文字列に変更された場合も対応しうるようマップコンテナを用いた．

3.1 基本アイテムの動作

採点プログラムにおける「基本アイテム」の役割は

1. 入力レコードから採点対象となるカラムに記述された文字列を取り出す
2. カラムに記述された文字列を参照し，記述内容と得点との関係を示した表から得点を求める

ことの 2 点である．

ここで，採点対象は同一のカラム幅を持つ 1 つ以上の（原則として）連続した領域の内容である．「基本アイテム」は，これらのカラムの内容を文字列ベクトルとして抽出する．対象となる領域は，開始カラム位置，カラム幅，繰り返し数の 3 つの整数によって指定される．

ここで，開始カラム位置が 1，カラム幅が 2，繰り返し数が 3 と指定されている場合を想定する．また，入力レコードの解答カラムに先頭から次のような解答が記述されているものとする．

```
----+----+
1234567890
```

このとき，読み取られる内容は第 1 要素が'12' であり，第 2 要素が'34'，第 3 要素が'56' の文字列ベクトルとなる．（ただし C++の STL ベクトルコンテナでは，第 i 要素にアクセスするための添え字は $i-1$ である．）

また，文字列ベクトルと得点の対の表をマップコンテナにより保持し，これを採点規則として利用し入力データに対応して得点を与える．もし，該当する文字列ベクトルが表になければゼロ点を与える．別解や部分点の付与は，この表に文字列ベクトルと得点との対を追加することによって，対応することができる．

さらに次の 2 つの属性を「基本アイテム」に定義し，これによって多様な採点方法に対応する．これらの属性は真偽いずれかの値を持つものとする．

これらのうち一つは「順不同性」であり，もう一つは「部分加点性」である．「順不同性」は，順序を問わない解答の採点に対応するためのものであり，これが真値である「基本アイテム」においては，採点対象となる文字列ベクトルを整列し，さらに重複を削除したうえで，マップコンテナのキーとの照合を行う．

一方「部分加点性」が真である場合には，まず入力された文字列ベクトルをキーとし，これによって採点を行う．もし，該当するキーが存在しない場合には，文字列ベクトルの個別要素のそれぞれについて表引きを行って採点し，各要素の得点の和を，この「基本アイテム」の得点として定義する．

これら 2 つの属性によって，かなり多様な採点パターンに対応することが可能となるが，現状では過去に蓄積された採点パラメータの記述との整合性を考慮し，すべての機能を採点に用いてはいない．組み合わせ採点において多く利用しているのは，「順不同性」による重複記述の削除および「部分加点性」の後半の性質である．

また，「部分加点性」が該当せずしかも順序を問わない採点については，上記の「順不同性」の指定にはよらず，別解（解答文字列の文字順を入れ替えたもの）の列挙によって対応している．

ここで採点規則の具体例を示す．

1. 最初の例として，数学の問題に現れる 3 カラムの数字記入を考える．1 カラムから 3 カラムが'-12' である場合が正解で 4 点を与え，'-24' には部分点 3 点を与えるものとする．このとき，「順不同性」および「部分加点性」は共に偽とし，入力カラムの指定は，開始カラムが 1，カラム幅が 3，繰り返し数は 1 である．また，採点規則を定義する表は最初の規則のキーが'-12' であり，得点が 4，第 2 の規則のキーが'-24' であり，得点が 3 と定義する．
2. 次に順不同の組み合わせ採点で，個別の解答に得点を与える例を検討する．ここで記入カラムが，1 カラムから 2 カラムであり，マークすべき文字が A と B であり，これらの記入順は問わず，しかも個別に 2 点，合計で 4 点が与えられるものとする．このとき，「順不同性」と「部分加点性」は共に真とし，開始カラムは 1，カラム幅は 1，繰り返し数は 2 とする．ここで採点表の第 1 規則は，キーが'A' で得点は 2，第 2 規則はキーが'B' で得点は 2 と定義しておく．

   受験者の解答が BA である場合には，読み取られる文字列ベクトルは ('B', 'A') となる．「順不同性」が真であるので，このベクトルは整列されて ('A', 'B') となる．次に，採点表をキー ('A', 'B') で検索するが該当するもの

がないため，次に個別の要素'A' を文字列ベクトルとみなして表を引くと得点 2 が得られる．さらに'B' についても同様に表を引くと得点 2 が得られ，合計 4 点がこの「基本アイテム」の得点となる．

次に受験者の解答が'AA' である場合を考える．読み取られる文字ベクトルは ('A', 'A') であるが,「順不同性」の指定により，重複要素が削除されるため ('A') が採点対象となる．これで表を引くと得点 2 が得られ，これが「基本アイテム」の得点となる．

3. もうひとつの若干複雑な順不同の組み合わせ採点，例えば，解答が'A' または'B' のいづれかのみである場合には 2 点であるが，両者の記入がある場合には 5 点を与える場合を考える．以下で,「基本アイテム」の機能を十分に用いる場合と，現状での処理指定の方法の両者を示す．

「基本アイテム」の機能を十分に用いる場合には，カラム指定は前項と同様に,「順不同性」および「部分加点性」の両者を真と置く．ただし，採点表の第 1 規則は ('A', 'B') をキーとし，5 点を得点とする．第 2 規則は'A' をキーとし，2 点を得点とし，さらに第 3 規則は'B' をキーとし，2 点を得点とする．この場合，解答カラムの記述が'AB' または'BA' である場合には，5 点が与えられ，一方のみが記述されている場合には，2 点となる．また'AA' および'BB' に対してもそれぞれ 2 点の採点結果が得られる．

上記のような指定を行うと，1 つの「基本アイテム」によってこの採点を実現することが可能であるが，現状ではこれまでのプロジェクトによって記述されてきた採点パラメータを用いており，これらの書式の性質から，上の指定方法採用することが困難であるため，実際には複数の「基本アイテム」によって一つの「採点単位」を構成する方法を用いた．

ここで 1 つ目の「基本アイテム」は前項で指定したものと同様とし，2 つ目の「基本アイテム」を開始カラムが 1，カラム長が 2，繰り返し数が 1 とする．2 つ目の「基本アイテム」の採点表の第 1 規則は，'AB' をキーとし，得点は 1 点，第 2 規則は'BA' をキーとし，得点は同じく 1 点とする．

これらの 2 つの「基本アイテム」を要素として持つ「採点単位」を設定すると，'A' あるいは'B' の記述のいずれかのみがある場合には 2 点が与えられ，'AB' あるいは'BA' の解答の場合には，1 つ目の「基本アイテム」の得点が 4 点であり，2 つ目の「基本アイテム」の得点が 1 点であるため，合計 5 点が与えられる．

ここで，前者の指定方法 (1 つの「基本アイテム」による処理）が困難であるのは，これまでにプロジェクトで記述されてきた採点パラメータが，指定された (開始カラム, カラム長, 繰り返し数) の各組について，特定の得点を与える文字列を一つ以上（複数個の指定は同一得点の別解をあらわす）指定する形式をとっているためである．これを利用すると，新たな形式での指定法を採用することが難しかった.「基本アイテム」の持つ構造には，採点表の一つの規則における文字列ベクトルによる記号の並びと，採点表の複数の行による並びの 2 重の繰り返しの構造が含まれている．前者の並びは，入力データをどのように定義するかにかかわるものであり，一方後者は入力と採点との関係を記述するものである．

前回プロジェクトからから引き継いだ採点パラメータの形式においては，繰り返しは基本的に別解を表すものであり，これは「基本アイテム」における採点表の規則の繰り返し構造に対応するが，文字列ベクトルの並びに直接相当するものがない．前回プロジェクトにおける採点プログラムにおいては，採点タイプの指定により，複数行にわたる採点方法の指定を，場合によって複数カラムの処理方法の指定として解釈していた．これを同一の「基本アイテム」へのパラメータ指定に反映させることも不可能ではないが，同一の形式のパラメータ記述が場合によって異なる構造として解釈されるのは，誤りを招きやすいと思われたため，より単純なパラメータの解釈方法を採用した．

上記の「基本アイテム」とこれらを組み合わせた「採点単位」の機能によってかなり多様な採点方法を記述できる．

ただし，文字列の個数が 3 個以上であって，採点が正解と一致する文字列の個数によって定まり，しかもその得点が正解との一致数には比例しない場合の処理（たとえば 3 個の解答のうち正解が 1

個ならば1点，2個なら4点，3個なら8点）は，上記の機能では簡潔に記述するのが困難である．実際，共通第1次学力試験における採点には，このような場合がある．これらに対応するために，3カラムまでのケースについては，採点パラメータの指定が冗長になったが，誤答カラムを含む解答パターンを列挙した．また4カラムを採点対象とする1つのケースについては，採点方式に特殊ケースを設定し，その採点処理を「基本アイテム」の特別なメンバー変数により表現した．

また，組み合わせ採点の対象となるカラムが連続していない場合もまれにあるため，カラム幅の指定が負数である場合には，カラム幅1を仮定しさらに指定された数値の絶対値によって指定される幅のカラムを読み飛ばすよう設定した．

さらに，同一のカラム範囲について2つの部分点を含む組み合わせ採点方式AとBがあり，方式Aによる得点と方式Bによる得点いずれか大きいものをこの項目の得点として定義する必要もあった．この場合A-Bに対応する「採点単位」およびBに対応する「採点単位」の2つを構成する．Aの得点がBよりも大きい場合には，1番目の「採点単位」の得点はA-Bであり，2番目の「採点単位」の得点はBであり，合計点はAとなる．また，Aの得点がBの得点よりも小さい場合には，A-Bは負となるため，1番目の「採点単位」の得点は零点となり，合計点はBとなる．

3.2 採点復元のためのパラメータ記述

過去に実施されたセンター試験の採点方法を復元するために，発表された正解表と成績ファイル中の情報から，各試験科目について表1に示す情報を整備した．

ここで，「採点タイプ」，「開始カラム」，「カラム幅」，および「繰り返し数」が同一である連続したパラメータ行は，同一の「基本アイテム」への指定とみなしている．また，「大問番号」と「箱番号」が同一である連続した「基本アイテム」は，同一の「採点単位」の構成要素であるとしている．このようにして構成された「採点単位」の通し番号を「項目番号」とし，大問あるいは科目における「採点単位」の数を「項目数」としている．このため，本システムにおける「項目数」は正解表に記述されている設問の「項目」とは一致しない．正解表においては，同一の「項目」として記述されていたとしても，それが複数の「採点単位」から構成される場合には，データベースへの収録においては複数の項目があるものとみなす．

3.3 選択問題の取り扱い

科目によっては，複数の大問から1つあるいは2つ以上を選んで解答する指示がある．平成2(1990)年から平成8(1997)年の間については，「数学II」において3つの大問から2問を選択することが指示されている．平成9(1998)年以降では「数学II・数学B」，「情報関係基礎」などにおいて大問選択がある．

昭和60(1985)年以降については，成績ファイル中に記述された大問選択にかかわる情報を利用し，これに基づいて採点処理を行った．

このうち平成2(1990)年から平成8(1996)年までの「数学II」においては，選択パターンにより同一のカラム位置に記述される解答が異なる大問に対応するため，事前に大問選択パターンを参照しつつフォーマット変換を行い，同一設問に対する解答が常に同一カラムに位置するよう事前処理を行ってから採点処理の対象とした．

また，平成9(1997)年の「情報関係基礎」においては，4つの大問（問）があるが，これらのうち，第1問と第2問が必須であり，第3問と第4問はいずれかを選択して解答する．ただし，第2問は3つの部分（A，B，C）に分かれており，受験者はこれらのうちいずれか一つを選択して解答する．現状の処理では，入れ子になった選択を記述することが難しいため，このケースに限り変則的に，第2問Aを大問番号2，第2問Bを大問番号3，第2問Cを大問番号4に対応させ，また第3問を大問番号5，第4問を大問番号6とみなして扱った．

また，昭和59(1984)年以前の共通第1次学力試験については，大問得点が記述されているものの，大問選択情報が明示的にファイル中には記述されていないため，各選択問題の得点を求め，高い得点の大問を採用し，同点の場合には番号の若い問題が選択されたものとした．

表 1 採点復元のための項目パラメータ

番号	パラメータ名称	データ型	内容
1	行番号	整数	記述されたパラメータの行番号 （プログラムにより再定義する）
2	問題文ファイル名	文字列	項目別試験問題 PDF ファイル名 （修飾子を除く）
3	箱番号	文字列	試験問題文における解答記入欄の名称「1」や「ア」など 組み合わせ採点により，一つの「採点単位」に複数のカラムが対応する場合には，「1-2」や「ア--イ」のように記述する
4	大問番号	整数	試験問題の大問番号（「問題 1」等と表記される） ただし平成 8 年の「情報関係基礎」は変則的な扱い
5	必須・選択の別	整数	1（必須）または 2（選択）
6	別解の数	整数	指定された得点を与える正解（あるいは部分点）の パターン数から 1 減じた数
7	配点	整数	指定された解答文字列に与える得点（部分点の場合もある）
8	採点タイプ	整数	「順不同性」,「部分加点性」（後述）等採点方法の指定 1：単独カラムの選択肢 2：複数カラムの組み合わせ, 3：解答順を問わず個別解答に部分点を与える（「順不同性」＋「部分加法性」のケース）．同一の「採点タイプ」,「開始カラム」,「カラム幅」，および「繰り返し数」を持つ複数行のパラメータを続けて指定する 4：欠番（旧「採点くん」では部分点のない順不同組み合わせ） 5：数字記入による組み合わせ解答（2 と処理方法は同一） 6：数字記入による順不同，部分点付加の組み合わせ解答（3 と処理方法は同一） 7：連続しないカラムの組み合わせ解答．この場合「カラム幅」は無条件に 1 を仮定する．入力パラメータの「カラム幅」欄には負の整数を指定し，この絶対値が読み飛ばすカラム数の指定となる（平成 2 年以降に該当なし） 8：欠番 9：無条件の正解 10–12：欠番 13：採点タイプ 3 の変形版．4 点の選択肢 4 つの順不同解答ただし 1 件正解で 4 点，2 件正解 6 点，3 件 10 点，4 件は 16 点
9	開始カラム	整数	解答欄の開始位置（解答欄の先頭を 1 とする）
10	カラム幅	整数	一つの解答文字列の長さ「採点タイプ」7 の場合は負の整数
11	繰り返し数	整数	解答欄の繰り返し数
12	正解文字列 1	文字列	指定の得点を与える文字列（部分点の場合もある）
13	正解文字列 2	文字列	同上（別解の数が 1 以上のとき）
14	正解文字列 3	文字列	同上（別解の数が 2 以上のとき）
15			以下同
16			...

4 統計量の導出

前回のプロジェクト（石塚他，2005）においては，試験，大問，項目，および選択肢の 4 つの水準で試験問題の解答結果にかかわる統計情報を求めた．今回も前回と同様に 4 つの水準で統計量を求めたが，表示内容は大幅に簡略化し，また項目反応モデルによる試験問題の特徴パラメータや，最適配点にかかわる推定値は省略した．これは，計算に時間がかかり，計算が正常に行われているかの確認を全ての科目について行うことが困難であることと，もう一つには前回用いられていた 2 パラメータの項目反応モデルが，次に述べるように必ずしもセンター試験の特性記述に適切ではないためである．

センター試験のかなりの設問においては，4 択あるは 5 択の形式で出題が行われるが，この場合にはもっとも下位の学力層での正解率がゼロにはならないため，2 パラメータモデルの仮定が適切ではない．しかしながら，チャンスレベルの正答率を指定することのできる 3 パラメータモデルの適用は，モデル識別性の問題が予想されるため，基礎資料として用いることが躊躇された．また，これまでに 2 パラメータモデルによって得られた係数を検討すると，著しく正答率の低い設問や，あるいは高い設問において，2 パラメータモデルの識別パラメータが見かけ上高く推定されることがあり，詳細なモデル診断を行わずに基礎資料としてこれらの推定値を採用することは，誤解を招く可能性があると思われた．

これらに代わる情報として，ここでは各設問項目（採点単位）の得点あるいは大問の得点と，それを除く設問項目（あるいは大問）得点の和との相関係数を，「得点相関」の名で示した．大問の場合，当然ながらそれ自身の得点を含める科目得点と除いた場合とで，相関は大きく変わるが，設問項目の場合にも，無視できない相関係数の違いが生じる．

今回の対象となった範囲の大問別得点の場合には，それぞれの大問得点を科目別得点に加えるか否かによって，平均で約 0.18 程度の違いがある．また，項目別の得点については平均で約 0.06 程度の違いがあり，0.1 程度の違いが生じる場合もある．このため，設問項目についても，その項目得点を除く科目得点との相関を資料として用いることにした．

また，受験者を科目得点の大小によりほぼ同数の 5 群に分割し，それぞれの受験者群別の各設問項目についての解答選択肢の選択率（数字記入など複数カラムの場合には，各解答パターンの記入率）を示した．ここで，大問の選択がある場合も，5 群の分割は選択の有無によらず同一の基準を用いている．大学入試センターにおける既存の資料（「設問解答率分析図」と呼ばれる）では，選択問題がある場合には，当該の大問を選択した受験者が同数の 5 群となるよう，選択問題ごとに分割の基準が変更されているが，今回収録したものは問題の選択には依存していない．このため，当該の問題を選択したものに限ると，各群の受験者数が等分から大きく離れる場合がある．

また，本プロジェクトにおいて収録した試験のうち，80 パーセント分位点が満点となるものがあった．つまり，受験者全体の 20 パーセント以上が満点を得ている．このケースについては，成績別の 5 群のうち，最上位群と上から 2 番目の群を併合した結果を示した．

5 統計情報閲覧システムの構築

部内のローカルネットワーク上で，Web ブラウザによって統計情報を容易に閲覧できるようにするため，上記の方法で導出された各種の統計量を Web サーバ上に各種の統計情報を記述した HTML ファイルを置き，またデータベースによる検索機能を一部併せることによって統計情報の閲覧システムを構築した．

石塚他 (2005) においては，データの閲覧を行うために，PC 上のデスクトップデータベースの機能を利用しているが，今回の開発では，基本的に HTML によって情報を表示する単純な方法をとった (図 1)．ただし，「設問解答率分析図」および「得点分布図」については，表示すべき図の件数が多くなるため（前者については収録した昭和 54(1979) 年から平成 20(2008) 年までの範囲で約 28,000 件），あらかじめ画像ファイルを用意しておくことが困難であるので，データベース上に保持された表示用データをスクリプト言語（PHP）を用いて参照

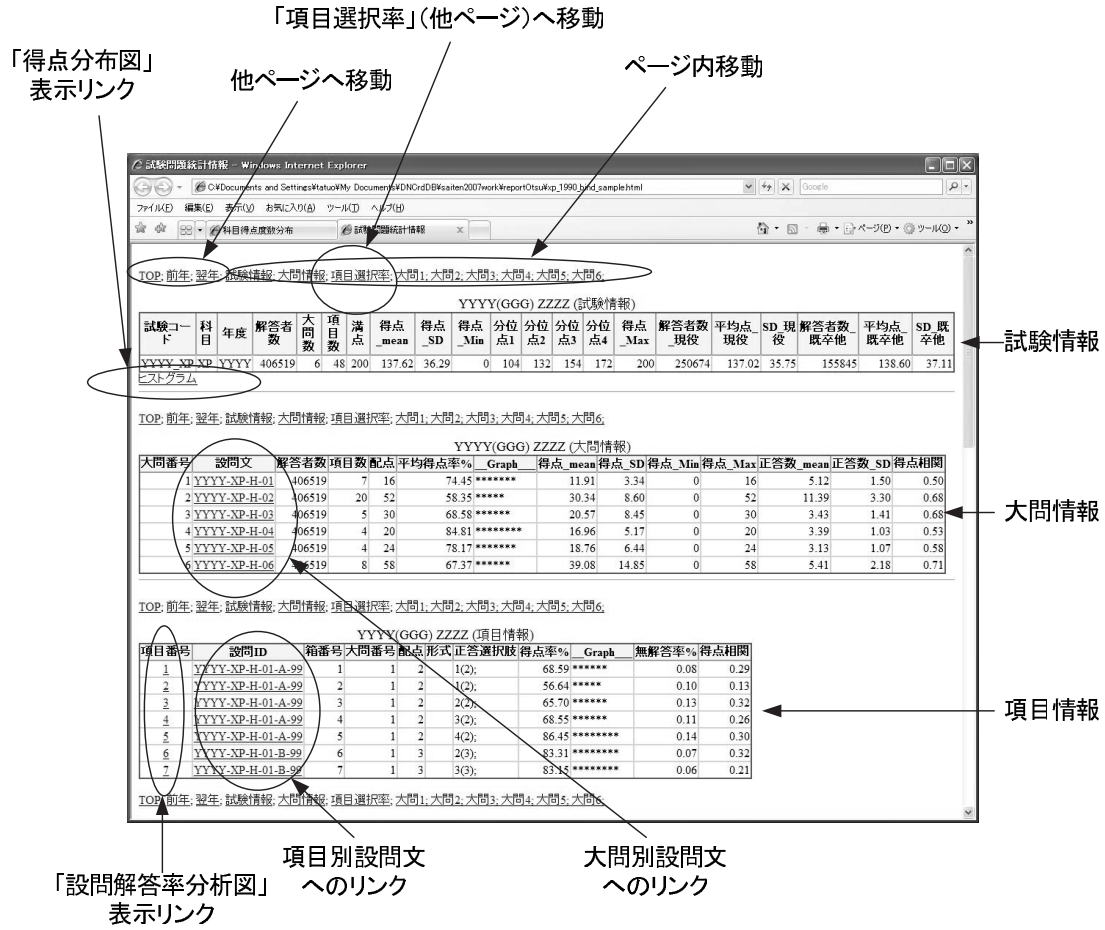

図 1 試験問題統計情報の表示例（デモ）

し，動的に図を生成する（図 2）.

また，各大問および設問項目に対応する試験問題ファイル（PDF ファイル）へのリンクを，統計情報とともに配置し，該当する試験問題を容易に参照可能とした．原則として，1 科目試験についての情報を，1 つの HTML にまとめる方針をとり，そのうちに試験情報，大問情報，および項目情報を各々表として示した.

HTML による表示を行うにあたっては，フィールド毎にタブコードで区切られた複数のテキストデータを読み込み，これらの情報を編集し，XHTML に整形して出力する Prolog プログラムを作成した．一般的には，このような作業を行うには DOM などの XML 処理プログラムを，C++や Java などの開発言語から利用するか，これらの XML 処理ライブラリを Perl，PHP などのスクリプト言語を通じて利用することが現在では多い．ここでは，幾分非標準的でなあるが，XML 文書の取り扱い能力が高い Prolog 上の XML パーザを利用することにより，XHTML 生成を実現した.

Prolog プログラムを実行する処理系としては SICStus Prolog 4.0 (The Intelligent Sytems Laboratory, 2007) を用い，John Fletcher によって Prolog によって記述された XML パーザを用いて XHTML を出力した (Fletcher, 2007) ．Fletcher による XML パーザは，SICStus Prolog によって第 3 版からライブラリとして一部改変されて採用されている．ここでは，SICStus Prolog によるライブリではなく Fletcher 自身による Prolog コードを原本とし，これに日本語の利用を容易にするための改変を一部加えて利用した．Prolog の文法やプログラミング技法については，Clocksin & Melish (2003)，および Sterling & Shapiro (1994) が詳細な解説を与えている．ただし，処理速度の効率化については，Prolog 処理系の特性により異なる対処が必要なため，それぞれ固有の性質を把握する

図 2 統計情報と試験問題の表示画面例

必要がある．

SICStus Prolog はスウェーデンコンピュータサイエンス研究所（SICS）によって開発された Prolog 処理系であるが，Prolog 処理系として最も広く利用されているもののひとつである．処理が高速であり，メモリ管理が柔軟で大規模な問題に適用しうること，および Unicode への対応がとられているため，日本語を含む多国語処理が容易であることなどの特徴がある．SICStus Prolog 4.0 では，実際 CPU クロックが 1.2GHz の Pentium M プロセッサで，760MB のメモリを持つノート PC を用いる場合，長さ 1 千万の整数リストの生成と反転（reverse）を，Prolog の作業領域（スタックとヒープ）がメモリ上に確保済みならば，約 2.5 秒で実現できる．また，Core2 Duo 6600（2.4 GHz）を搭載したデスクトップ PC の場合には，同様の処理を約 0.8 秒で完了できる．SICStus Prolog は文字列を Unicode 文字番号の値から構成される整数のリストとして扱うため，多バイト文字の取り扱い方法が一貫している．ただし，第 4 版では，第 3 版でサポートされていた EUC などへの対応を廃止し，Unicode のみをサポートするようになった．

Windows 版でのコンソール（DOS Window）を対象とする標準入出力においては，マルチバイト文字がロケールを考慮して変換して入出力されるため，問題なく日本語を利用できる．ただし標準入出力を用いる場合であっても，ファイルが入出力先となる場合には，ローカルコードではなく Unicode による入出力が行われる．ファイルに記述された日本語文字がすべて Unicode のみであれば問題はないが，Windows 上ではここで処理対象となる既存資料が Shift JIS（より正確には CP932）によってエンコードされたテキストであるため，事前変換が必要になり作業が煩雑である．このため Shift JIS エンコードテキストをバイナリデータとして読み込み，Unicode との変換をおこなう関数を C 言語で作成し，これを SICStus Prolog の処理系に組み込み，入力用の述語として利用した．

5.1 XHTML の生成手順

XHTML 作成のための具体的な手順は次のようなものである．

1. Shift JIS で記述された統計情報のタブ区切りファイルをバイナリーモードで入力し，これ

```
/* 区切り記号 (タブコード) を述語 (fact) で定義 */
tabfields_delimiter_code(9).

/* 複数フィールドの定義 */
text_fields([F|Fs])     -->
        text_field(F),tabfields_delimiter_code, !,
        text_fields(Fs).
text_fields([F])        --> text_field(F),!.
text_fields([])         -->  [].

/* 個別フィールドの定義 (非タブ記号) */
text_field(F)           -->
        tabfields_nondelimiter_codes(Cs),!,
        {atom_codes(F,Cs)}.
text_field([])          --> [].

/* タブ区切り記号の定義 (DCG) */
tabfields_delimiter_code --> [X],
        { tabfields_delimiter_code(X)}.

/* 非区切り記号文字列の定義 */
tabfields_nondelimiter_codes([C|Cs]) -->
tabfields_nondelimiter_code(C),
        tabfields_nondelimiter_codes(Cs),!.
tabfields_nondelimiter_codes([]) --> [].

/* 非区切り記号の定義 (\+ は述語の否定を示す) */
tabfields_nondelimiter_code(C) -->
        [C], {\+ tabfields_delimiter_code(C)}.
```

図 3 確定節文法（DCG）によるタブ区切りデータの定義

を Windows のライブラリ関数を用いて変換し Unicode の整数リストを得る．SICStus Prolog には，書式付入力の機能が用意されていないため，タブ区切りデータの読み込みの機能を実現するプログラムを作成する必要がある．ここでは，DCG（確定節文法）と呼ばれる文法記述機能を用いて，テキストデータの解釈方法を記述することにより，書式付データ入力を実現した．図 3 に DCG によるタブ区切りテキスト行の処理プログラムの一部を示す．ここでの記述は-->演算子の左辺が，右辺に定義される下位要素から構成されることを示しており，中括弧{ }で区切られた部分は，文法定義を補助する条件の記述にあたる．DCG の記法と機能については，Clocksin & Mellish (2003)，あるいは Sterling & Shapiro (1994) などの Prolog の標準的教科書に記述がある．

ここで，text_fields_str などの記述は，Prolog の処理系によって差分リストを用いた Prolog の述語（真偽値を値としてもつ関数）に変換される．この述語は，文字列をあらわす文字コード整数のリストを入力し，これが，下位要素 text_field と tabfields_deimite_code，および text_fields の 3 者から構成されているか（最初の記述による指定，2 つ以上のフィールドが存在する場合），あるいは text_field から構成されるか（1 個のフィールドの場合），または空リスト（0 個のフィールドの）場合であるか，これら 3 つの条件のいずれかが成立するならば真であり，処理が継続される．一方，これらいずれの条件も成立しなければ，述語は偽となり，text_fields_str としての解釈が失敗する．

図 3 に示したものは，DCG 利用の単純な例であるが，Fletcher による XML パーザも類似のプログラミングテクニックを用いて記

```
xmltemplate(sample1, [Attributes,Body],
            xml( [version="1.0", encoding="UTF-8"],
                 [element(html,Attributes,Body)]
                 )
            ).
```

図 4 Prolog で記述された XML テンプレートの例

テンプレートへの代入と変換

```
| ?- xmltemplate(sample1, [[attr1="AA",attr2="BB"],
                [element(elem1,[],[pcdata("XXX")])]], XMLdoc),
   xml:xml_parse(Codes,XMLdoc),
   format("~s",[Codes]).
```

作成された XML 文書

```
<?xml version="1.0" encoding="UTF-8"?>

<html attr1="AA" attr2="BB">
 <elem1>XXX</elem1>
</html>
```

図 5 テンプレートへの代入と XML 文書への変換

表 2 収録した主な統計量

水準	統計量の種類（部分）
試験情報	受験者人数，科目別の平均点，科目得点標準偏差，最高点，最低点，20%刻みの分位点 現役生の平均点，それ以外の受験者の平均得点など
大問情報	選択者数，大問別平均点，得点率，得点標準偏差，得点相関
項目情報	得点率，無解答率（解答欄空白の率），得点相関
項目選択率	科目得点 5 群別の解答選択肢の選択率

述されている (Fletcher, 2007)．DCG を用いて入力処理プログラムを記述することは，処理効率の面からは必ずしも得策ではないが，かなり入り組んだ構造をもつデータやスクリプトを解釈するプログラムを容易に記述できる利点がある．現在では，PC ハードウェと Prolog 処理系の性能向上により数万件のデータも必要なメモリが実装されていれば問題なく処理できる．

2. 各大問あるいは設問項目と問題文 PDF ファイルとの対応関係を調べておき，統計情報の一覧中に適切な URL を記述し，閲覧時に参照を可能とする．
3. 設問解答率分析図を参照するために，統計一覧表の指定の箇所に，描画処理を行う PHP スクリプトへのリンクを記述する．試験コード（年と科目のための識別記号）と，項目番号（箱番号とは異なる通し番号）をスクリプトへの引数として記述しておく．
4. あらかじめ XHTML で作成した表示例を用い，これを Fletcher による XML パーザを用いて Prolog の内部形式（複合項）へと変換し，その一部を変数に置き換えることによって表示用のテンプレートを作成する．このテンプレートの指定箇所に，上記の手続きによって作成されたデータを当てはめることにより，表示用の表に対応する複合項を作成する．次に，XML パーザの逆向きの処理（Prolog のパターンマッチング機能により，複合項から XML 文書への変換もパーザを用いて可能になる）を行い，この複合項を XML 文書の文字列に変換する．この結果をファイルに出力する．

図 4 は，簡単なテンプレートの例であり，図 5 はこれの変数部分に代入を行い，XML 文書に変換後，文字列として表示したものであ

る．実際の作表では，操作はより複雑になるが，基本的にはこのようなテンプレートへの代入により，簡潔に文法的に正しい XHTML 文書を作成することができる．

これらの設定を行うことにより，関連情報を容易に参照するシステムを作成できる．

5.2 設問解答率分析図の生成

設問解答率分析図とは，各設問項目について，問題文中に示された選択肢がどのような頻度で選択されているかについての情報を示すものである．ここでは，科目得点により受験者をほぼ同数の 5 群に類別し，各学力階層別の選択率を示すことにした．

項目別の統計情報などの導出と同時に，各選択肢の選択頻度を求め，これに基づき選択率を計算する．ただし，組み合わせ配点のある場合，および数学などにおける複数カラムへの数字選択の場合には，選択パターンがかなり多くなる場合がある．これらの場合には，低頻度の選択パターンが多数列挙される可能性があるため，採点・集計プログラムを用いた統計量の出力の際，および設問解答率分析図の表示の際に一定の基準を設け，表示パターンを制約した．

採点・集計プログラムによる選択頻度の統計情報出力にあたっては，配点のある選択肢については全ての解答パターンを出力するが，配点のない（ゼロ点）場合については，全体での解答率が 10 パーセントを超えるものに限定した．また，設問解答率分析図の表示にあたっては，配点のある場合について全体での解答率が 5%以上，あるいは成績が最も良い群での解答率が 5%以上のいずれかの場合に表示を限定した．組み合わせ解答の場合には，個別カラムの解答頻度は表示せず，組み合わせパターンの頻度を表示する．図 2 に表示例を示す．

また，選択問題の存在する科目については，科目成績による 5 群類別は科目受験者全体に基づいて行う．このため，ある大問を選択した受験者の成績が全体にくらべ著しく偏っている場合には，特定の群に該当する受験者が少数になる場合もある．もし，層別された特定の群の該当者が存在しない場合には，便宜的にいずれの選択肢の選択率もゼロと定義する．また，特に成績最上位群の該当者が存在しない場合には，第 4 群と最上位群とを合併して表示する．大学入試センターにおけるこれまでの資料作成の慣例では，選択問題のある場合について受験者の類別を科目受験者全体ではなく，その問題を選択した受験者に限定して行っている．ここでは，この慣例から違反することになるが，複数の図の比較検討を行う場合，分類基準が一貫していることがより望ましいと判断されるため，このような方針をとった．

設問解答率分析図の描画は，次のような方法で実現した．

1. 各設問項目（採点単位）について，解答パターンの選択率を求め，上記の基準を満たすものをタブ区切りレコードとして出力する．
2. 試験コード（実施年と科目コードを合わせた情報を含む）および項目番号（採点単位の通し番号）をキーとして，関係型データベース（MySQL4.1.7 以降）に登録する．この際データベースの内部文字コードは UTF-8 に設定する．
3. 統計情報を表示する XHTML 文書中の，項目別の統計情報の表示箇所に，前項のデータベースを検索する PHP スクリプトへのリンクおよび試験コードと項目番号をスクリプトへの引数として記述する．
4. 試験コードと項目番号をキーとしてデータベースを検索し，選択率データを得る PHP スクリプトを Web サーバに用意する．さらに折れ線グラフを描画するスクリプトも共に用意し，これを用いて設問解答率分析図を描画する．

現状では，Web サーバとして Windows 2003 Server 上の IIS を用い，データベースには，MySQL AB 社による MySQL4.1 あるいは MySQL5.0 を用いている．さらにデータ検索用のスクリプト言語 PHP 第 5 版を利用し，PHP による描画ライブラリとして JpGraph（2 版以降，Aditus Consulting Inc. 2007）を利用した．これらのソフトウェアのインストールに当たって注意すべき点を以下に示す．

1. MySQL は 4.1.7 版以降を利用し，内部文字コードは PHP などでの日本語処理の整合性を考慮し UTF-8 に設定する．
2. PHP5 を Web サーバから CGI として起動で

きるよう設定する．PHP5 に標準で用意されている MySQL インタフェースライブラリには mysql と mysqli とがある．機能は後発である mysqli の方が高いが，本プロジェクトの開始時において安定的であった mysql モジュールを現状では利用している．

3. PHP の配布ソフトウェアに同梱されている mysql モジュールの版が，サーバにインストールした MySQL のバージョンと合わない場合があるため，これらの整合性をとることに注意する．MySQL AB 社から，サーバソフトウェアと適合する PHP の拡張ライブラリが配布されているので，これを利用することができる．また，多バイト文字処理関連の拡張ライブラリ mbstring，および描画のための拡張ライブラリ gd2 が利用できるよう設定する．
4. 図に日本語を表示する必要があるため，JpGraph の PHP のスクリプトを，UTF-8 エンコーディングによって記述しているが，このとき PHP のプログラムファイルの先頭に BOM（Byte Order Mark，Unicode でのエンコーディング方式をプログラムが認識するための文字）が存在すると，生成された画像データがこの BOM とともにブラウザに送信される．この場合，ブラウザはデータをテキストファイルとして認識してしまうため，図の表示ができない．現状では，図は PNG フォーマットで出力するよう設定してある．Windows のテキストエディタ（notepad を含む）では，通常 UTF-8 エンコーディングを用いてテキストファイルを編集すると，保存時に BOM が付与されるため，PHP プログラムの編集を行う際に注意が必要になる．
5. JpGraph の初期状態では，日本語フォントの指定がされていないため，JpGraph のシステム設定ファイルにて適切なアウトラインフォントを指定する．

5.3 全文検索システムの利用

統計情報の一覧中には，大問および項目別の試験問題文（PDF ファイル）へのリンクを設けてあるため，統計情報から試験問題の参照は容易であるが，実用上はむしろキーワードによって試験問題を検索し，これに該当する統計情報を参照する必要が大きい．

この機能を実現するためには，試験問題文の全文機能が必要になる．文書検索についてはインターネット検索の隆盛とともに，多くのツールが開発されている．本プロジェクトにおいては，商用の文書検索システム（ビジネスサーチテクノロジ社製の WiSE）を導入し，この機能を一部ローカライズすることにより，試験問題文から該当する統計情報の参照を研究開発部内 LAN において可能にした．

WiSE は LAN あるいはインターネット上の文書を検索対象とし，これらについての n-gram インデックス（構造化された文字列索引）を構築することにより，文書検索サービスを Web サーバ上で提供するものである．インデックスの構築は，あらかじめ設定されたスケジュールにしたがって深夜などにクローラと通称されるプログラムが検索対象ファイルを走査して構築する．検索サービスは HTTP サービスとして利用者に提供される．利用者が検索キーワードを入力すると，キーワードを含む対象文書のリストが表示される．

一般的に，文書検索システムにはキーワード方式（有意味な語彙リストを保持し，それらの所在を示すもの）と，n-gram 方式（任意の文字のつながりのパターンについてインデックスを保持するもの）とがある．ここで採用した WiSE は，n-gram 方式をとる．それぞれ長所短所があるが，n-gram の長所はキーワード語彙の管理を必要としないこと，記号列など語彙として認識させることが難しい文字列の検索も行えることなどがある．一方，欠点はインデックスが巨大なものになり，しばしば検索対象の文書よりも大きくなってしまうことがある．ここでは，検索対象はセンター試験問題文に限定しているため，さほどインデックスが大きくならないこと，記号列や複数の外国語についての検索を行う必要があるため，n-gram 方式を採用することにした．

これらの文書検索システムが提供するサービスは，通常キーワードの存在する文書を示し，URL をリンクとして提供することにあるが，ここでは表示スクリプトの一部を改変し，該当する文書に関連する統計情報一覧を示す文書の URL も同時に表示することとした．これにより，キーワードを指定することにより，試験問題文の検索とそれ

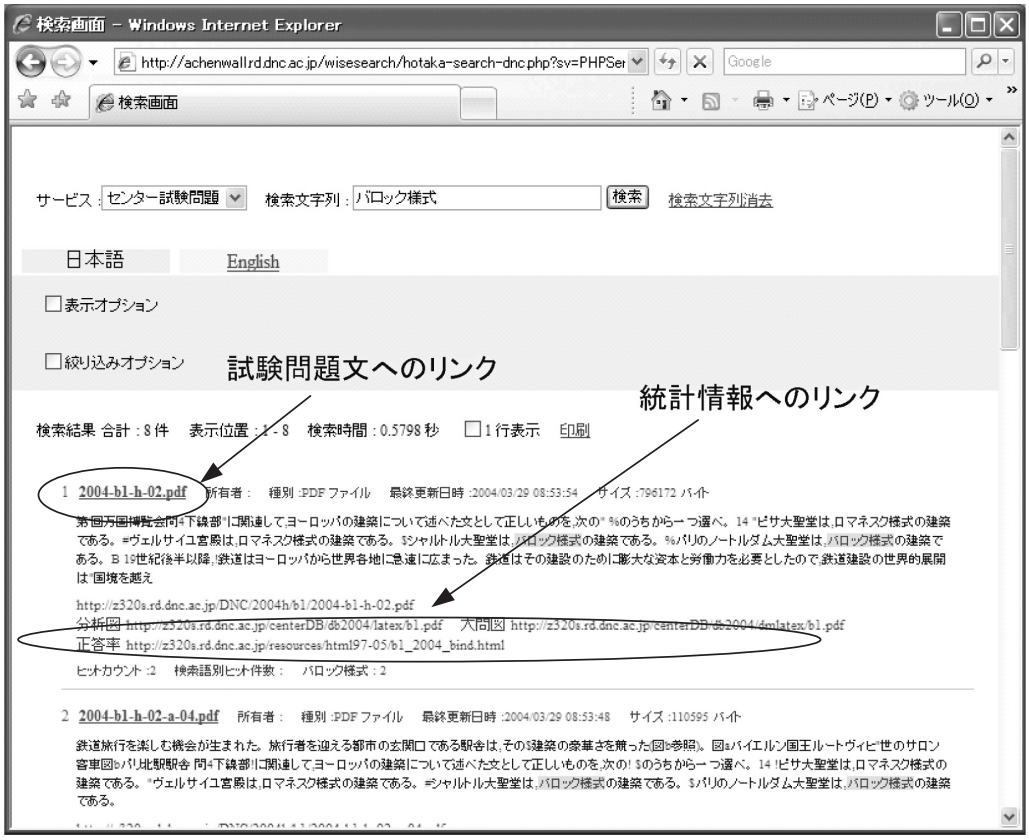

図 6 試験問題の検索画面
（「バロック様式」の文字列を含む試験問題の検索結果の表示例）

に対応する統計情報を容易に参照することが可能になった．図 6 は「バロック様式」との検索語を入力し，これを含む試験問題文を表示した画面である．図の上方に検索語の入力フォームがあり，ここに検索すべき文字列を入力すると，該当する試験問題文のファイル名が列挙される．また，統計情報を含むページへのリンクを同時に表示するよう WiSE の処理スクリプト（PHP で記述されている）を一部変更した．

かつては，文書検索システムは企業の基幹システムでの利用を念頭において販売されており，数百万円以上のライセンス費用を必要とするものがほとんどであったが，平成 17 年ころから低価格の商品が販売されるようになった．WiSE のほかに低価格で利用可能な検索システムとしては Google Mini（Google 社の提供する検索サーバの小型のもの），Microsoft Search Server などがある．また，最近では，Hyper Estraier や RAST など日本国内で開発され無償で公開されている n-gram ベースの文書検索ソフトウェアもある．また，一般的な商用のシステムでは，かなり高額であるが XML データベースと文書検索機能の両者を提供するシステムも現れている．

一般的な業務支援のための情報システムにおいては，各種の機能を提供するために，階層的なメニュー構造をもつシステムを用意することが多いが，このような汎用的な検索システムを利用すると，このような詳細なメニューをあらかじめ用意することなしに，利用者が必要な情報にたどり着くことを可能にできる．

試験問題にかかわる情報環境を整備するには，これらの文書検索の技術動向は重要であり，今後も研究と業務の改善に関連性の高い技術や製品が現われるものと推測される．

6 過去データにみる設問の統計的特徴

ここまでは主に，大学入試センター試験の実施

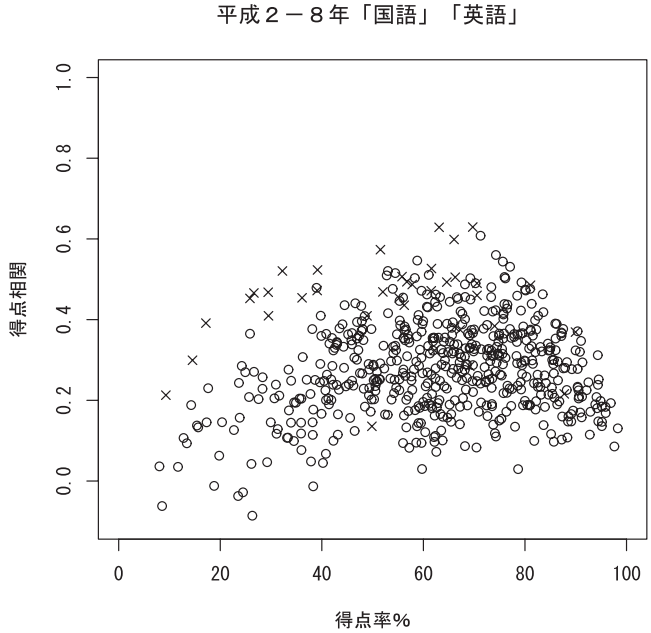

図 7 「国語」と「英語」の項目別統計量（平成 2 年–平成 8 年）
円は単一カラムの項目，X は複数カラムの採点単位（項目）を示す．

によって得られた成績データに基づく試験問題統計情報の整備について，技術的な解説を加えることに主眼を置いてきた．最後にこれらを通じて得られた試験問題の統計的特徴について簡単に紹介する．

中等教育の状況との関連など，これらの情報についての広い視野からの検討が行われるべきであるが，これらの視点に基づく分析は多くの研究者によってこれらの情報が共有され，共通の認識基盤の上に立っての議論が広く行わなければ難しい．ここでは，現在知りうる範囲での統計的な特徴に限って検討を加える．現在までに整備を行った試験統計情報は，昭和 2(1979) 年から平成 20(2008) 年までのものであるが，以下ではこのうち平成 2(1990) 年から平成 8(1996) 年の範囲の設問項目について検討する．

図 7 は，この範囲の「国語」および「英語」の各項目（採点単位）について，正答率（得点率）と得点相関との散布図である．円で表示されているのは，単一の解答カラムの項目であり，X で表示されているのは複数の解答カラムを持つ項目である．正答率が極めて低い場合，あるいは高い場合には得点相関が多くの場合，ゼロに近くなっていることがわかる．図 8 には，「国語」と「英語」のそれぞれについて，正答率の 2 次関数を説明変数とし得点相関を従属変数とする回帰分析による条件付き平均の推定値を示した．上側の曲線が「英語」を示し，下側が「国語」を示す．

平成 9(1997) 年以降においては，「国語 I」および「国語 I・国語 II」の科目得点の大学への提供が大問別に行われており，一部の大学においては「古文」・「漢文」の得点を利用しない場合があるが，ここで分析対象とした 7 年間においては，大問別の得点提供を行っていない．このため，「古文」・「漢文」の無回答率が特に高くなることはなく，設問項目別に検討すると最大の無回答率でも 2 パーセントを超えない．ただし，平成 9 年以降においては，「古文」・「漢文」の無回答率は次第に増加している．

一方，図 9 は，同じ範囲の「数学 I」および「数学 II」についての散布図である．こちらも正答率が 100 パーセントに近い場合には，やはり得点相関がゼロに近づくが，正答率が低い場合には，前者とは幾分様相が異なり，得点相関がゼロに近づかないものが多く存在する．図 10 は同様の回帰曲線である．ここでは 2 つの科目をまとめて分析した．

「国語」および「英語」の設問においては，ほとん

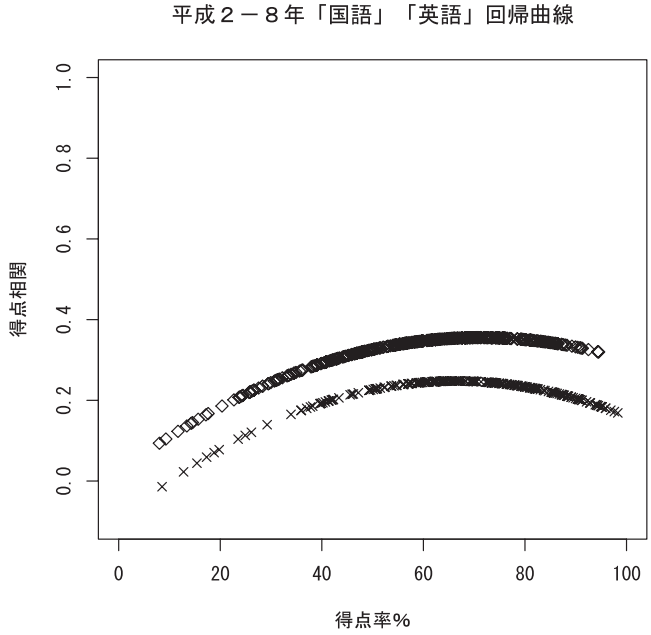

図 8 「国語」と「英語」の項目別統計量（平成 2 年–平成 8 年）の回帰曲線
2 次関数による推定値．上側◇が「英語」，下側 X が「国語」．

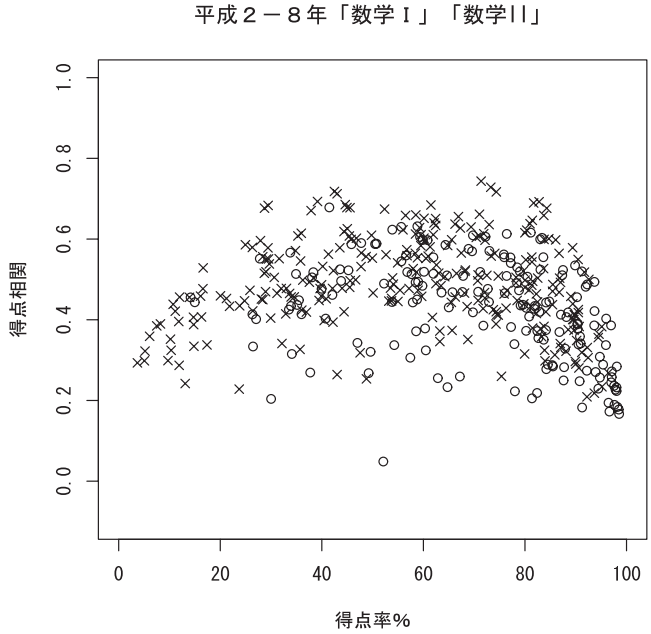

図 9 「数学 I」と「数学 II」の項目別統計量（平成 2 年–平成 8 年）
円は単一カラムの項目，X は複数カラムの採点単位（項目）を示す．

どの項目が単純な多肢選択式の解答方式であるため，難問の場合にはチャンスレベルによる正答の影響が大きくなる．このため，正答することが難しい項目における得点の分散のほとんどの部分が，あて推量による正誤の分散に基づくものとなり，科目得点との相関に寄与しなくなる．これは，いわばSN 比の悪い信号を得ていることに相当する．

一方，「数学 I」および「数学 II」においては，数字を複数のカラムに指定するなどの，組み合わせ方式による採点を行う項目が多い．複数のカラムへの記述によって採点が行われる場合には，チャンスレベルによる正解の可能性は大幅に減少する

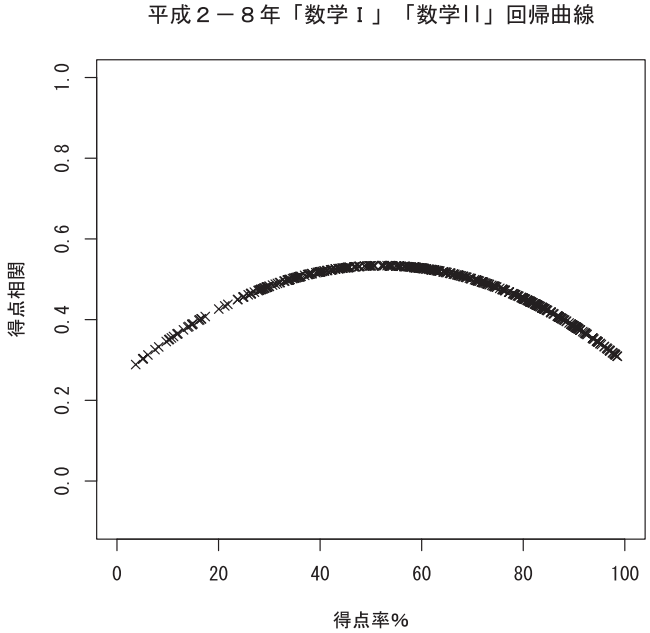

図 10 「数学 I」と「数学 II」の項目別統計量（平成 2 年–平成 8 年）の回帰曲線 2 次関数による推定値.

ため，たとえ難問であっても，それが科目学力を良く反映するのならば，科目得点との相関は，必ずしも小さくはならない．また，単一の解答カラムの項目についても，多くの場合前後の解答との関連が深いため，他の項目と独立した解答が得られることが少ないと推測される．これら 2 つの図に見られる得点相関の傾向の違いは，このような理由に基づくものと考えられる．

実際，平成 2(1990) 年から平成 8(1996) 年までの本試験について検討すると，得点率が 20 パーセント未満であり，なおかつ得点相関が 0.4 より大きい設問項目は，14 存在するが，これらはいずれも数学関連の科目のものである．このうちには，単一のカラムへの解答によって得点を与える項目も 4 つ存在するが（図 8 にはこれらのうちの 2 項目が表示されている），これらにおいてはいずれも正解よりも多くの頻度で解答を得ている「おとり」の選択肢（数値）が存在しているため，チャンスレベルによる正答率が 10 パーセント（0 から 9 まで数値をランダムに選択する場合の正答率）よりさらに小さくなっている．

さらに，広くテストデータの分析に利用されている項目反応理論（IRT）に基づくモデルを用いてこれらの特徴について簡単な検討を加える．

まず，能力パラメータ θ が固定されている場合の正答率を慣例に従って次の式で表す (Lord, 1980)．ここで Y は受験者が正解したならば 1，誤りの解答ならば 0 となる 2 値の確率変数である．

$$\Pr(Y=1|\theta) = c + \frac{1-c}{1+\exp\left(-1.7a(\theta-b)\right)} \tag{1}$$

ここで a は識別力パラメータ，b は難度パラメータ，c はあて推量の確率（受験者がまったく知識を持たなくとも偶然に正解する確率）を示すパラメータである．

さらに能力パラメータ θ が標準正規分布に従うことを仮定する．ここで ϕ は標準正規分布の確率密度関数であるとすると，受験者全体における正解率 μ は次の式で表される．

$$\mu = \Pr(Y=1) = \int_{-\infty}^{\infty} \Pr(Y=1|\theta)\phi(\theta)d\theta \tag{2}$$

ここで，a, b, c の 3 個のパラメータの値が変化するとき，2 値変数 Y と潜在的な能力変数 θ との相関係数がどのような値になるかを検討すると表 3 のような値が得られる．計算には R-2.5 (R-project core team, 2007) の積分関数（`integrate`）を用い，θ の積分範囲を -4 から 4 までとして求めた．

表 3 3 パラメータ項目反応モデルにおける得点相関

パラメタ値			正答率	正答者平均	正答者 SD	相関
a	b	c	μ	$E(\theta\|Y=1)$	$\sigma(\theta\|Y=1)$	$r_{Y\theta}$
0.5	−2	0	0.82	0.14	0.96	0.30
1.0	−2	0	0.92	0.11	0.94	0.37
2.0	−2	0	0.96	0.08	0.94	0.38
0.5	−1	0	0.68	0.24	0.94	0.35
1.0	−1	0	0.76	0.29	0.88	0.51
2.0	−1	0	0.81	0.29	0.83	0.61
0.5	0	0	0.50	0.37	0.92	0.37
1.0	0	0	0.50	0.56	0.83	0.56
2.0	0	0	0.50	0.71	0.71	0.71
0.5	1	0	0.32	0.50	0.93	0.35
1.0	1	0	0.24	0.90	0.81	0.51
2.0	1	0	0.19	1.27	0.63	0.61
0.5	2	0	0.18	0.63	0.94	0.30
1.0	2	0	0.08	1.23	0.84	0.37
2.0	2	0	0.04	1.88	0.61	0.38
0.5	−2	0.25	0.86	0.10	0.97	0.25
1.0	−2	0.25	0.94	0.08	0.96	0.32
2.0	−2	0.25	0.97	0.06	0.95	0.33
0.5	−1	0.25	0.76	0.16	0.97	0.28
1.0	−1	0.25	0.82	0.20	0.93	0.42
2.0	−1	0.25	0.86	0.21	0.89	0.51
0.5	0	0.25	0.62	0.22	0.97	0.28
1.0	0	0.25	0.62	0.34	0.94	0.44
2.0	0	0.25	0.62	0.42	0.90	0.55
0.5	1	0.25	0.49	0.25	1.00	0.24
1.0	1	0.25	0.43	0.38	1.03	0.33
2.0	1	0.25	0.39	0.46	1.07	0.37
0.5	2	0.25	0.39	0.22	1.02	0.18
1.0	2	0.25	0.31	0.24	1.09	0.16
2.0	2	0.25	0.27	0.17	1.13	0.11

ここで，表の最初の 3 つの列は項目反応モデルのパラメータである．4 列目は受験者全体における正解率 μ を示し，5 列目は正解者における θ の期待値であり，6 列目は正解者における θ の標準偏差である．7 列目は 2 値変数 Y と θ との積率相関係数 $r_{Y\theta}$ である．試験統計情報に示した得点相関は，該当する項目の得点を除いた得点和とその項目の得点（多くは 2 値）との相関 r_{YZ} を示したものであり，厳密にここでのとの相関 $r_{Y\theta}$ に対応はしない．それぞれの Y と θ との相関が一定であることを仮定すると，Z は他の Y の誤差成分を含むため，r_{YZ} は $r_{Y\theta}$ より小さめの値になる．

ここで p を設問の個数とし，各設問の正誤を表す変数 $Y_j, j = 1, \ldots, p$ が次の式で表されるものと仮定しよう．

$$Y_j = \alpha\theta + \beta\varepsilon_j \tag{3}$$

式中の α と β はそれぞれ定数であり，ε_j は θ および他の $\varepsilon_{j'}$ と独立な成分を表す確率変数であり，分散は 1 とする．また α および β は正の数とする．ここで

$$\beta = \sqrt{1-\alpha^2}$$

と仮定する．このとき Y_j の分散は 1 であり，Y_j と

θ との相関は，次の式で表される．

$$r_{Y_j\theta} = \frac{\alpha}{\sqrt{\alpha^2+\beta^2}} = \alpha \tag{4}$$

さらに α および β が全ての j について同一であると仮定し，$p-1$ 個の変数の和を次のように定める．

$$Z_j = \sum_{i\neq j} Y_i \tag{5}$$

このとき Z_j と Y_j との相関係数 $r_{Y_jZ_j}$ は次の式で与えられる．

$$r_{Y_jZ_j} = \frac{\alpha^2}{\sqrt{\alpha^2+\beta^2/(p-1)}} = \frac{\alpha}{\sqrt{1+\frac{\beta^2}{\alpha^2(p-1)}}} \tag{6}$$

一般的に $r_{Y_j\theta}$ より $r_{Y_jZ_j}$ のほうがゼロにより近い値になる．たとえば $\alpha = 0.3$ および $p = 30$ とすると，ほぼ $r_{Y_jZ_j} = 0.258$ であり，また $\alpha = 0.4$ の場合にはほぼ $r_{Y_jZ_j} = 0.368$ となる．式の形から p が大きくなると，Z_j の分散における θ に寄与する分散の比率が大きくなるため，この違いは小さくなる．

表 3 では，識別力をあらわす a パラメータの値が 0.5, 1，および 2 の 3 通り，また難易度を表す b パラメータが -2, -1, 0, 1，および 2 の 5 通り，またあて推量での正解確率を示す c パラメータが 0 および 0.25 の 2 通り，あわせて 30 ケースの数値を示した．

3 番目の c パラメータがゼロの場合には，表に示した中では $a = 2, b = 0$ のケースにおいて，$r_{Y\theta}$ の値が最も大きい 0.71 となっている．また b のいずれのケースについても，識別力 a の値が大きいほど $r_{Y\theta}$ は大きくなっている．一方，3 番目のパラメータ c が 0.25 の場合（選択バイアスがないと仮定するならば 4 択問題に相当する）では，$b = 0$ のとき，識別力パラメータ a が大きいと $r_{Y\theta}$ が大きくなる．しかし，問題が難しく正解率が低い $b = 2$ の場合には，他のケースとは逆に，a の値が大きいと逆に得点相関が小さくなっている．これは，前述の通り c パラメータがゼロではない場合には，問題が難しいと正誤を示す確率変数の変動部分の多くがあて推量によるものとなり，学力を識別するための指標としての性能が悪くなることを示している．

以上，過去の試験データおよび理論的な検討の両者から，多肢選択形式の設問においては，少なくとも合計得点を用いた合否判断をする限り，難問は学力を評価するためのツールとしてはうまく機能しないことが分かる．正誤 Y_j と潜在的な学力 θ の相関は，正解率が $c+(1-c)/2 = c/2+1/2$ ほどのとき（$c = 0.25$ ならば 0.625）最も高くなりうることがわかる．このため，設問の正答率が 6 割程度であるべきという要請は，試験のあり方についての理念的な要求である以前に，多肢選択形式をとる試験がうまく機能するための技術的要件であることが分かる．

7 まとめと課題

簡潔な統計情報を集約し，ブラウザで容易に閲覧可能とすることにより，過去に実施された試験問題の特性についての検討を支援することができる．第 6 節に示したように，これらの情報を詳細に検討することにより，試験問題の統計的な特性をより明らかになった．これらの情報を十分に活用することにより，試験問題の妥当性，信頼性の向上に寄与すると共に，新たに作成される試験問題の品質の向上に寄与することが期待できる．

現在までに，昭和 54 年からの共通第 1 次学力試験と平成 2 年以降の大学入試センター試験について本試験の統計情報を平成 20 年実施分まで整備した．過去の採点方式を振り返ると，共通第 1 次学力試験においては，かなり複雑な採点方式がみられるが，平成 2 年以降には複雑な採点方式はほとんど見られない．これは，関係者によって作業の困難さが共通に認識されて，標準的な採点方式の採用が問題作成において一般化したものと推測される．しかしながら，このような標準化にいたるまでに 10 年近い時間が経過していることを考えると，作業の利害得失についての共通の認識を関係者が共有するまでには，かなりの時間がかかるものであると考えなければならない．

成績データの収録範囲の拡大は，現状のデータ整備の枠組み内で継続的に対応できるが，入学試験の社会的役割を考えるとより広い範囲での研究の継続を考える必要がある．

試験にかかわるより進んだ知見を得るには，教科科目内容，試験問題，および受験者の解答特性の 3 者の関連を，統合して検討する必要があると

思われる．このためには教科科目の内容をなんらかの形式で電子的に組織化する必要が生じる．関係する情報としては，各学問領域に固有な知識，高校での履修範囲，および学習者がどのように知識を獲得し運用するかについての知識などが含まれる．これらの情報を組織化は，統計情報の整備と比較して，格段に難しい課題であり，研究方法についての大きな革新が必要になる．しかしながら，研究の水準を高めるためには，これらの課題への取り組みは今後不可避と思われる．

今回のプロジェクトでは扱わなかったもう一つの点は，統計情報のデモグラフィックな特性である．現役生（卒業見込みの高校生）とそれ以外の受験生の平均点は示したが，それ以外の属性別の集計は示していない．地域別，高校学科別，性別などの情報の取り扱いにはかなり慎重にならざるを得ないが，社会科学的あるいは行政施策上これらの情報が重要な知見を与える可能性も大きいため，解答パターンのみならずより広範囲な統計情報との統合を準備する必要が考えられる．

謝　辞

データベースの整備作業にあたっては，小野塚暁子，國見敦子，桑原由起子，山内久美の皆さんに大変お世話になりました．心から感謝いたします．

参考文献

石塚智一 ・大津起夫・鈴木規夫・石岡恒憲・吉村 宰・荘島宏二郎・橋本貴充・中畝菜穂子 (2005)「平成 14 年度～16 年度 共同研究 I 報告書 試験問題統計情報の整備に関する研究」, 大学入試センター 研究開発部, 東京：大学入試センター.

大津起夫 (2004) 統計メタデータ記述の技術動向と XML 文書の利用, 大学入試センター研究紀要, No.33, 65–88. 東京：大学入試センター.

Aditus Consulting Inc. (2007). *JpGraph 2.3*, http://www.aditus.nu/

Clocksin,W.F. & Mellish,C.S. (2003) *Programming in Prolog: Using the ISO standard, 5th ed.*, Springer.

Fletcher, J. (2007). *This Prolog Line: John Fletchr's home on the web.* http://www.zen37763.zen.co.uk

Lord, F.M. (1980). *Applications of item response theory to practical testing problems*, Hillsdale, New Jersey: Lawrence Erlbaum.

PHP manual project (2007). *PHP manual*, http://jp.php.net

R Development Core Team (2007). *R: A language and environment for statistical computing*, Vienna, Austria: R Foundataion for Stattistical Computing.

Sterling, L. & Shapiro, E. (1994). *The art of Prolog, 2nd ed.*, MIT press.

The Intelligent Systems Laboratory (2007). *SICStus Prolog User's Manual release 4.0.1*, Kista Sweden: Swedish Institute of Computer Science.

Statistical information databases of the NCT items and development of their usage environment

OTSU Tatsuo*
ISHIOKA Tsunenori*
HASHIMOTO Takamitsu*

Abstract

The National Center Test (NCT) is a nationwide university entrance examination organized annually by the National Center for University Entrance Examination (NCUEE) and Japanese universities. All national and public universities in Japan have been adopting the NCT. In addition, many private universities use the NCT as one of the tests to screen students. In January 2008, about a half million participants took the NCT. Although the item development procedures of the NCT is a highly comprehensive process involving reviews by anonymous professors, the NCUEE does not use modern test theory to moderate difficulties. Organizing statistical data of the NCT is key to item writing for future examinations and educational policy making. We developed a database of item statistics of the NCT between 1979 and 2008. This database contains several statistics based on classical test theory. We included about 28,000 items from various areas. The data is contained in XHTML forms, which are generated using Prolog based XML parser, with links to scripts for dynamic chart generation. The forms also contain links to documents for reference. The statistics reveal difficulties in moderating scores in the absence of pre-test operations.

Key words: database, item difficulty, discriminating power, text retrieval

* Department of Test Analysis and Evaluation, Research Division
The National Center for University Entrance Examinations