A-3

情報技術学習支援システムの開発と学習評価
－Random Forest を用いた問題文の分類－
Development of the Information Technology Learning Supporting System and Learning Evaluation
- The classification of problem statement by field using the Random Forest –

○宮川　裕介[1]，泉　　隆[2]
*Yusuke Miyakawa[1], Takashi Izumi[2]

Abstract: In this study, we are considering how to improve the learning motivation of the learner. For this purpose, it is necessary to classify the problem statements to provide an appropriate problem for the level of the learner. We propose a method that can be classified by field the problem statement using the Random Forest is a kind of machine learning.

１．はじめに

インターネットを用いて学習を行う e-Learning システムが教育機関や企業の研修で多く利用されている．e-Learning システムを利用した学習での欠点として，学習の頻度が学習者の学習意欲に依存することが挙げられる．学習意欲が低下することは，目標とする試験や達成度に大きな影響を与える．そのため学習意欲が向上するシステムが求められる．

本研究では，個人の特性に合わせたテストを作成し，その結果をフィードバックすることで，学習者の自己調整を助け，学習意欲の向上をはかることを目的としている．そこで，e-Learning システムで提供する問題を分野ごとに分類することで，学習者が苦手とする分野やまだ勉強していない分野を出題しないといった個人の特性を考慮したテストを生成できると考える．本報告では，機械学習の一種である Random Forest を用いた問題文の分野別分類について検討した．

２．問題文の分類

本システムが対象とするのは基本情報技術者試験である．本試験は午前と午後に分かれており，午前に出題される問題は，大分類, 中分類, 小分類に分けられている．そして，最も細分化されている小分類は 100 の分野によって構成されている．これらの各分野は解くのに必要とされる知識項目が異なるため，問題文を分野ごとに分類することで学習者の苦手な知識項目を詳細に把握することができると考える．しかし，人手による分類はコストやミスが生じるため，Random Forest を用いて問題文の属する分野を推定する手法を検討する．

２．１．Random Forest

Random Forest は複数の決定木を用い，識別を行う機械学習アルゴリズムである．Random Forest に用いられる個々の決定木は高い識別性能をもたないが，それらを複数用いてそれぞれの結果を補うことによって高い予測性能を得るアンサンブル学習ができるのが特徴である．

２．２．Random Forest の手順

本研究での Random Forest の手順について述べる．

①　文字列の抽出

本報告では問題文およびその正解選択肢を形態素解析エンジン MeCab により解析する．そして，解析により抽出された形態素の有無を入力データとして用いる．

ここでは抽出する文字列として以下の 3 種類を抽出する．そして，文字列の種類によって分類精度にどのような変化があるのか確認する．また，C)の特徴語とは対象試験のシラバスに記載されているキーワードである．

A)　全ての形態素の文字列
B)　品詞が名詞である形態素のみの文字列
C)　特徴語と同じ品詞パターンをもつ文字列

②　ブートストラップサンプルによるデータ集合の生成

ブートストラップサンプルとは N 個のサンプル集合の中から重複を許してランダムに標本を選んでできた新しいサンプル集合のことを指す．本報告では手順①ですべての問題から抽出した文字列より d 次元のデータ行列を生成する．そして，各問題における文字列の有無$\{x_i\}$をデータ行列 X によって表現し，それと分野 C_i のセットを学習データとしている．そして，これら学習データの集合からブートストラップサンプルにより学習に用いるデータ集合を生成する．

１：日大理工・院（前）・情報　２：日大理工・教員・情報

$$D_L = \{(X_1, C_1), (X_2, C_2), \cdots (X_N, C_K)\}$$
$$\mathrm{X} = \{x_i\}_{i=1}^{d} \ , \ \ \Omega = \{C_i\}_{i=1}^{K} \qquad (1)$$

D_L：学習データ集合，Ω：分野の集合

C_i：各分野，$x_i \epsilon \{0,1\}$：文字列の有無

X：文字列の有無を特徴とした問題文のデータ行列

③　決定木による学習

決定木の各ノード t において式(2)に示すジニ係数より学習に用いるデータ集合 D_L を 2 つの集合に分割する．ジニ係数は[0,1]の値をとり，値が大きいほど分割結果がばらついていることを示す．ジニ係数 I_G が最も小さくなるパラメータを推定し，入力データに対して{left, right}を返す分岐関数を$\{x_i\}$ごとに作成する．そして，$\{x_i\}$から d'個ランダムに選択する． これを終了条件が満たされるまで続け，決定木を生成する．

$$I_G(t) = \sum_{i=1}^{K} P(C_i|t)\left(1 - P(C_i|t)\right) \qquad (2)$$

$P(C_i|t)$：ノード t においてクラス Ci のデータが選ばれる確率

④　複数の決定木による識別

手順②，③より複数の決定木を生成し，図 1 に示す Random Forest を構築する．これら決定木の出力はクラス C_i となっており，出力したクラスの多数決によって入力データの分類先が決定される．

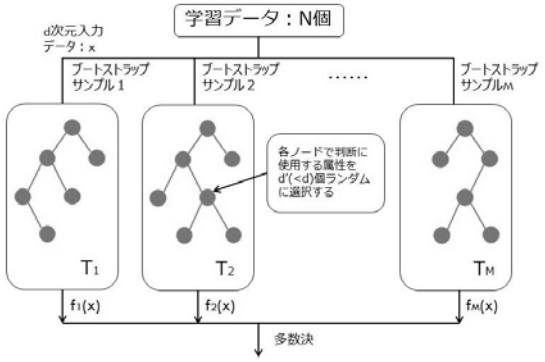

Fig1. Configuration of the discriminator by Random Forest [2]

3．実験

Random Forest を用いて問題文の分野別分類実験を行った．実験に用いる学習データやテストデータは表 1 の通りである．なお，分類正解率は全テストデータのうち，分野別分類に正解したテストデータ数を表しており，式(3)のように定義する．また，本実験では Python の機械学習ライブラリである scikit-learn[3]を用いた．

$$(\text{分類正解率}) = \frac{(\text{正解したテストデータ数})}{(\text{テストデータ数})} \times 100\ [\%] \qquad (3)$$

Table1. Experimental condition

学習データ	平成 20～23 年度　春期・秋期 基本情報技術者試験 午前問題　640 問
テストデータ	平成 24 年度　春期 基本情報技術者試験 午前問題　80 問
対象文字列	A) 全ての形態素の文字列 B) 品詞が名詞である形態素のみの文字列 C) 特徴語と同じ品詞パターンをもつ文字列
測定項目	分類正解率

Table2. Classification rate string A, B, and C

	小分類	中分類	大分類
分野数	100	23	9
A) 形態素［%］	61.25 (49)	72.50 (58)	78.75 (63)
B)　名詞［%］	66.25 (53)	77.50 (62)	82.50 (66)
C) 特徴語［%］	65.00 (52)	72.50 (58)	78.75 (63)

実験結果を表 2 に示す．なお，表 2 における括弧内の数字は正しく分類できた問題数を表す．表 2 より本実験で特徴に使用した 3 種類の文字列のうち，名詞が大中小分類すべてで最も高い値となり，小分類の分類正解率は 66.25[%]であった. 名詞が最も高くなった要因として識別に悪影響を与える形態素を選択する割合が減ったためではないかと考えられる．また，小分類の分類精度は大分類に対し，13~17[%]程度低下したが，これは使用した学習データの分野別の分布が均一でなかったことにより誤分類を抑制できなかったためであると考える．

4．まとめ

本報告では Random Forest を用いて, 問題文の分野別分類について検討した．その結果，名詞の文字列を特徴量とした場合, 100 の分野に対する正分類率は 66.25[%]と良好な結果であった. 今後は Random Forest に使用するパラメータや学習データの見直しを行い，さらなる精度向上を目指す.

5．参考文献

[1] 瀬沼航太郎，宮川裕介，泉隆：「情報技術学習支援システムの開発と学習評価―解答要因の推定―」，情報科学技術フォーラム講演論文集，Vol.12，No.4，pp.489-490，2013-08-20.

[2] 平井有：「はじめてのパターン認識」，森北出版株式会社，pp.175-197，2012-07-24.

[3] scikit-learn: machine learning in Python : http://scikit-learn.org/stable ， 2014-09