Detecting Student Misuse of Intelligent Tutoring Systems

R.S. Baker, A.T. Corbett,　and K.R. Koedinger,

in Proc. Intelligent Tutoring Systems, 2004, pp.531-540.

1 Introduction

- 頻繁にシステムを不適切に利用する生徒は予備知識やありふれた学力を制御し適切にシステムを利用した学生の 2/3 ほどしか学習をしなかった。それゆえ生徒のモチベーションと認知に反応することのできるシステムが存在することであれば現行システムよりも効果的である。(Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. 2004)

- システムをゲームする(Baker, R.S., Corbett, A.T., Koedinger, K.R., Wagner, A.Z. 2004)
  - 課題を達成すためにシステムを利用するにあたって特性や規則性をシステマチックに利点を得ることで教育的タスクを良いパフォーマンスにしようとする狙いの行動の事である。

- 生徒は 2 つのゲーミングに従事する。

Help abuse と Systematic trial and error

- ゲーミングの頻度は学習に対して強い負の相関が存在するが、他の off-task な振る舞いの頻度は関連しないことを発見した。

- なぜ生徒がシステムをゲームするのかを理解することは、システムが生徒にどのように反応すべきかを決定するための要素になる。

- 本論文
  - 低い学習に関連するゲーミングをより成功的に気づける LRM の機械学習を示し議論。交差検定はモデルが効果的であることを示す。加えて、モデルは 2004 の Blaker らの仮説であるゲームする生徒はかなり難しいステップでしがちであるという仮説とコンビネーションさせる。

2 Methods

2.1data souece

- ゲーミング検査器のアルゴリズムを開発するために、生徒のパフォーマンスと振る舞いを coginitve tutor の散布図を用いた授業のデータからまとめた。

- 70 人の 1 グループ
  - それぞれの生徒は 71～478 の行動を tutor で行った。
  - 3 種類のデータを取得する
    - １つ目のデータ：それぞれのアクションに対して、２４の特徴抽出を行った。別ページに記載
    - ２つ目のデータ：生徒の授業中の振る舞いを人間の手によってコーディング。生徒がゲーミングに費やした時間割合の近似値 $G_0...G_{69}$
    - ３つ目のデータ：学習の結果を利用。

- 3 つのセットに生徒を分割。
  1. ５３人の生徒はゲーミングなし
  2. ９人の生徒は pre-post テストで高い点数を獲得し、ゲーミングによって悪化はしていない
     - **GAMED-NOT-HURT**
  3. ８人の生徒はあきらかにゲーミングで悪かった
     - **GAMED-HURT**
- GAMED-HUR と GAMED-NOT-HURT を区別することは重要である。
  - また GAMED-HURT にシステムによる介在の対象とすることもより重要である。

2.2 Data Modeling

- ３つのデータ資源を用いて、ゲーミング頻度の密度推定器を訓練した。
  - Latent Resopose Model の１つのセットにおける前進選択(Ramsey, F.L., Schafer, D.W.,1997)

- 参考：http://case.f7.ems.okayama-u.ac.jp/statedu/hbw2-book/node14.html

- 有効なパラメーターは以下から取り出す。
  - ２４の特徴(parameter*feture)として１次効果
  - ２次効果(parameter*feature^2)
  - 相互作用(parameter*featureA*featureB)

- モデル選別の間、パラメーター候補値のもっともベストな値を見つける
  - 反復勾配降下法を用いて我々のモデルの予測と実データとの平均絶対偏差をもっとも削減するものを加算された。
  - 前進選択は平均絶対偏差を適切に削減するのが見つからないまで実行。
  - 結果：最適合モデルは 4 つのパラメーターを所有６パラ以上のモデルは存在しなかった。

- 明確なモデルが与えられたら、そのアルゴリズムは最初に生徒がゲーミングをしたかどうかを予測する。
- 全ての生徒とすべての行動に *n* 個のパラメーターの１セットが与えられたら
  - 特徴 *Xi*(X_i^2 , orX_iY_i)に関連図けられた α、予測値 *Pm*（*m* の行動はゲーミングかどうかを
    $P_m = \alpha_0 X_0 + \alpha_2 X_2 + \alpha_3 X_3 + \cdots \alpha_n X_n$ を計算し求める）
  - それぞれの P_m はステップ関数を用いた閾値処理されている。
    ($P_m \le 0.5$ なら $P'_m = 0$、それ以外は $P'_m = 1$)
    - ✧ Tutor 内でのそれぞれの行動に対する P’m とい分類が与えられる
    - ✧ ゲーミングとして分類される生徒の行動予測 P’m の割合を決定することで、$G'_0 \cdots G'_{69}$ の値セットを与える
    - ✧ $G_0 \cdots G_{69}$ と $G'_0 \cdots G'_{69}$ を比較することで、オリジナルデータからの
    - ✧ ベストなモデルパラメーター発見のために、偏差は反復勾配降下とモデル選択の間用いられる。

- すべてのデータセットに対してベストモデルを発見するに伴って、モデルがオリジナルデータに存在しない生徒にどれくらい効果的に一般化できるか（将来的には異なる tutor の授業で用いるから）について LOOCV を求める
  - 参考：http://www.singularpoint.org/blog/r/model-selection-aic-vs-loocv/（モデル選択の実験）

LOOCV の間　70 人の生徒のうち 69 人のセットに適合させ、70 人の生徒に対してモデルが生成した予測値がどれくらい良かったのかを調査した。

2・3 Classfier

- ゲーミングをしている生徒を識別する分類機を開発すること。
  - 閾値を設定し、閾値よりも高い生徒はゲーミングをしたと認識され、それ以外はゲーミングをしていないと判断。
  - 正しくゲーミングを認識（hits）とゲーミングしていない生徒をゲーミングしていると判断 (false positives)

- 結果：ROC カーブを図 1 に示す。
  - 分類機の識別能力を A‘値で評価する。これはモデルがあるゲーミング生徒とゲーミングしていない生徒がいた時にそれらを正しく認識できる確率である(Donaldson,W,1993)

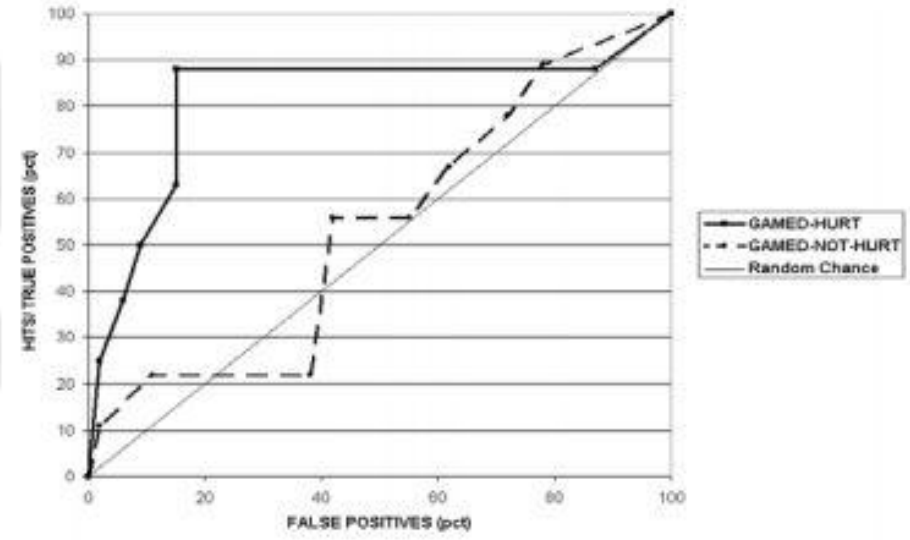

Fig. 1. Empirical ROC Curves showing the trade-off between true positives and false positives, for the cross-validated model trained on both groups of gaming students.

Results

3.1 Our Classifier's Ability Detect Gaminng Students（分類器の能力について）

- GAMED-HURT と GAMED-NOT-HURT の両方をモデルが扱えるように訓練させたなら、GAMED-HURT の生徒をゲーミングしていると必然に分類する(A'=0.82,p<0.001)

- hits と false positives との間に高い比率の閾値があるとき、分類器は GMAED-HURT の生徒のうち 88%をゲーミングとして正しく識別した。この間ゲーミングをしていない生徒の 15%をゲーミングと分類。
  - GAMED-HURT の生徒に対して介入を行う信頼アリ。

- GAMED-NOT-HURT の生徒をゲーミングと必然的に分類はしなかった(A'=0.57,p=0.58)

- GAMED-HURT を検出することが重要であるため、GMAED-HURT の学生のみにおいて訓練しているモデルから高い成果が生まれることが考えられる
  - しかし、GAMED-HURT の生徒のみで訓練したモデル(A'=0.77)は両方のグループで訓練されたモデルよりも分類性能が悪かった。

- ゲーミングはネガティブにポストテストスコアに影響していることが重要であるが、分類器だけでは学習し損なう事を分類しない。
  - モデルは低いポストテストの生徒または元々成績の悪い生徒を必然的に分類していない。
  - 要するに単にゲーミングをしているすべての生徒を識別しているのではく、生成の悪いすべての生徒を識別する。

3.2Describing Our Model

- モデルは生徒それぞれの行動がゲーミングであるかどうかを予測する。
  - モデルの予測がゲーミングについて何を示唆しているかを調査し、それらの予測はゲーミングをよりよく理解させることを支援している。
  - 表 1 に示すものが 0.5 以上のとき、行動がゲーミングであるとモデルは予測する。

Table 1. How the model predicts whether a specific action is an instance of gaming

Name	Coefficient	Feature
F_0: "ERROR-NOW, MANY-ERRORS-EACH-PROBLEM"	-0.0375	pknow-direct * number of errors the student has made on this problem step (across all problems)
F_1: "QUICK-ACTIONS-AFTER-ERROR"	+ 0.094	pknow-direct * time taken, in SD above (+) or below (-) the mean time for all students, on this problem step (across all problems)
F_2: "MANY-ERRORS-EACH-PROBLEM-POPUP"	+ 0.231	number wrong on this problem step (across all problems), if the problem step uses a popup menu
F_3: "SLIPS-ARE-NOT-GAMING"	- 0.225	pknow * how many errors the student made on last 5 actions

- **特徴 F0="ERRO-NOW,MANY-ERRORS-EACH-PROBLEM"（問題ごとに多くのエラー,今もエラーをしてる）**
  - 生徒がすでに少なくとも 1 つのエラーを現在の問題内の現在の問題ステップで起こして、たくさんのエラーを過去の問題においてのこの問題ステップで起こしているならばゲーミングしがちであると認識。
  - 生徒が過去においてこの問題ステップで沢山のエラーをだしたが、現在はそれをおそらく理解しているであろう（かつこの問題において間違ったステップをまだしていない）ならばゲーミングはしにくいだろうと識別する。

- **特徴 F1="QUICK-ACTIONS-AFTER-ERRO" （エラーの後の迅速な行動）**
  - 現在の問題で 1 つ以上エラーを出しおり、とても早い行動を現在しているならばゲーミングする傾向があると判断
  - 現在の問題のステップで 1 つはエラーを出しているが、後に続く行動はゆっくりであるとゲーミングをしにくい。
  - またはすでに良く知っているスキルを利用する初めての機会において早い解答をしているならばゲーミングはしにくい

- **F2"MANY-ERROS-EACH-PROBLEM-POPUP"**
  - 問題ステップにポップアップメニューが踏まれ複数の問題にわたって沢山のエラーをすることはゲーミングをさらに示すものとなる。
  - 反応が単独としては冗長である場合ポップアップメニューは複数の質問選択を利用している。が、すばやい継続においてそれぞれの解答を試みることを生徒にたいして可能にさせているのである。

- **F3"SLIPS-ARE-NOT-GAMING"（うっかりミスはゲーミングじゃない）**
  - 既存のスキルを高い確率で持っているならば、最近多くのエラーを起こしていてもその生徒はゲーミングをしにくい。
  - F0 と F1 のように生徒がすでに多くのエラーを最近の問題内の問題ステップにおいて起こしているならば、F0 と F1 は貧しいスキルと既知のスキルを区別することはないという事実に反している。

- 交差検定中に生成された 70 のモデルはとても高い類似性を生み出している。
  - F0,F2,F3 は交差検定されたモデルの 97%で見られたし、F1 は 71%で見られた。
  - 他の特徴は 10%以上を超えたのはなかった。

- モデルの驚くべき側面は生徒がヘルプを利用する特徴はなかったことである。
  - 筆者らは Help Abuse に関連した研究を行っている。にもかかわらずモデルは help abuse を直接検出することなくゲーミングを正確に検出していることは興味深い。

3.3 Futher Investigations With Our Model

- モデルの予測した 21520 のゲーミングの予測の 49%は、ゲーミングであったもっとも近い 4 つの行動のうち最低 2 つが存在するクラスタ内で起きた。

- そのようなクラスタの偶然的な頻度を決定するために、われわれはモンテカルロシミュレーションを行った。
  - それぞれの生徒のゲーミングと予測された事例がランダムに生徒の行動の 71 から 478 に分散
  - ゲーミング予測のうち 5%のみが、そのようなクラスタで起きた。それゆえにモデルは偶然的に予期されるよりも実質的によりゲーミング行動が起こることを示した。

- モデルはいつ GAMED-HURT と GMAED-NOT-HURT の生徒がゲームすることを選択するときの大きな違いが少なくとも 1 つ存在することをしめしている。
  - この違いはなぜ GAMED-GURT の学生は学習がより少ないであるのかを説明する

- スキルの難易度におけるゲーミング頻度の比較
  - “難しいスキル“におけるゲーミングの頻度、生徒が 20 %以下の chance of knowing を持っている（20%は授業開始直後に生徒がスキルを知っている確率）
  - “簡単なスキル”におけるゲーミング頻度（生徒は 90%以上の chance of knowing であること）
  - 結果：GMAED-HURT グループの生徒は難しいスキルにおいてかんたんなスキルもよりゲームするということをモデルは予測した。(t(7)=2.99,p<0.05)
  - 結果：GAMED-NOT-HURT グループの生徒は難しいスキルで簡単なスキルよりもゲームする時間が大幅には異ならないということをモデルは予測した。(t(8)=1.69,P0.13)
  - GAMED-HURT の生徒は彼らが損をするときにゲームすることを選択してしまう

4 Future Work and Conclusions

- システムをゲームし、乏しい学習を示す生徒を理解するのにモデルは成功的であった。

- 将来的に 3 つの目標がある。

1. 他の中学数学においてこの現象を研究する。そしてそれらの tutor へ我々の分類器を生成する。
   - ✧ 他の tutor でゲーミングの観測を集め、分類器をそれらの tutor でゲーミングを認識するか適応を試みる。
   - ✧ (Baker, R.S. Corbett, A.T, Koedinger, K.R, Wagner, A.Z,2004)での help abuse についての最近の予測にむけての生徒のゲーミングについてのモデルの予測の比較は追加的な洞察と機会を提供するだろう。
2. モデルは生徒がゲーミングしているとき正確に識別できるかどうかを確定的に決定すること
   - ✧ ラベルされたデータを集めることは、ログファイルにおける行動に対してのそれぞれの観測の正確な時間にリンクできる場合、このゴールに対して我々をアシストするだろう
3. どの生徒がゲーミングを削減するために介入を受けるか対してこのモデルを利用する
   - ✧ なぜゲーミングをするのか調査不足である。
   - ✧ ゲーミングへの適切な反応を設計することはなぜゲーミングをするのかを理解することを要求する。

- 本研究の長い目での目標は生徒の知識や認知的特徴に適応するだけでなく、生徒の振る舞い的特徴にも適応することである。そうすることで我々は tutor をより効果的な学習環境として作れる。

ログデータから抽出した特徴

＊ステップとは問題解決を前進させる解答のこと。行動(action は)ステップを進めるための行動。

・tutoring ソフトウェアの行動評価：正しい行動、incorrect&indicating の既知バグ（手続き的誤解）、incorrect だが indicating ではない既知バグ、またはヘルプの要求（バイナリ型の３変数）

・行動を伴ったウィジェットのインターフェースの種類：プルダウンメニュー、文字列、数字の入力、点のプロット,チェックボックスの選択（バイナリ型４変数）

・tutor の評価、過去の行動、生徒が現在の行動において既知スキルが含まれている可能性“pknow”とよぶ（ベイジアン知識トレーシングアルゴリズムを用いている(Corbett, A.T. & Anderson, J.R,1995)）

・この問題ステップにおいて生徒のはじめの試みは解答であったか？（またはヘルプを取得した？）

・”Pknow-direct” tutor のログファイルから直接取得（過去の２つの特徴はそれから抽出）。
もし問題ステップにおける正しい行動が最初の試みならば、Pknow-direct は pknow と等しい.
すでに問題ステップにおいて試みをしているのならば Pknow-direct は-1 である
Pknow-direct はよく知っているスキルと最初の試みと、より最近における試みとの対比をする

・どれくらいの時間を行動に用いたか

・最後の 3 または 5 ステップにどれくらいの時間を費やしたか（２変数）

・このスキルを練習する各々の機会においてどれくらい時間を費やしたか。問題全体にわたって平均する。

・全ての問題に対してこのステップの間違いを生徒が費やした総時間。（１つの問題内における複数の試みを含む）

・このスキルにおけるヘルプまたはエラーの時間、過去の問題らも含む

・この問題ステップに最後の５アクションは何回含むか

・最後の８アクションにおいて何回ヘルプを要求したか？

・最後の５アクションで何回エラーを起こしたか