# Development of an Intelligent Tutoring System using mixture modelling of learning curves and tendency distributions

Tiago Quintas

January 18, 2024

**Abstract**

The web application SIACUA is a plataform created to assist calculus students for an autonomous learning experience. To enhance the students' learning capabilites, over the years a new Intelligent Tutoring System (ITS) was being implemented, to provide guidance and recommendations to the user. In this paper we present an ITS capable of drawing learning curves for each student individually and for the global dataset, while recommending various study tips in real time. Mixture modelling and bayesian inference were used to draw the learning curves, as well as $t$-tests for the ITS's decision patterns, taking advantage of the bayesian networks already implemented in SIACUA. Tendency distributions were also developed and implemented to calculate the probability of succeeding and following tasks suggested by the ITS. Notifications and feedbacks were also prompted and suggested using rewarding methods and behaviour policies. The ITS was implemented in C$\sharp$, as an independent Class Library, to be used as reference for other web applications.

## 1 Introduction

Web-based intelligent tutoring systems (ITSs) have currently been widely used across many plataforms, to provide guidance and useful recommendations to students, while adapting to the student's learning characteristics. Learners with different backgrounds and learning capabilities already have available online knowledge resources in many web applications, but, for the teaching techniques to be successful, the ITS must draw accurate learning curves and practice schedules for each user individually, through a simple, dynamic and non-intrusive interface.

The application SIACUA (Sistema Interactivo de Aprendizagem por Computador da Universidade de Aveiro[1]) provides free online studying material, mainly for calculus students, but extended to all users. A regular user, after registering or loging in with the university's email, may solve exercices with multiple choice questions, study theory about searched topics, check his/her stats and query external learning resources. A Bayesian network is used to map prerequisite links between topics and to compute its knowledge beliefs. Each user has an instance of the network and everytime he/she answers a question the beliefs are recalculated. Solving exercices is the main feature of SIACUA, as it provides continuous updates on new exercices and a step-by-step solution for each one, improving the learning experience for the students.

The goal for the ITS is to enhance the students' learning capabilites, modelling each their learning path and adapt to its schedule routines. Taking advantage of the calculated beliefs from the Bayesian network, the ITS uses general mixture modelling of learning curves that map the probability of failing exercices of a given difficulty. Based on those paths, it then places the curve within the dataset and computes its empirical frequency. It also computes the learning curve within the user's dataset, modelling the primary, *lucky* and *unlucky* components. A new distribution is also used, the Tendency distribution, to model the pracitce routines and the probability of succeeding or following the suggested tasks. The ITS should compute possible thresholds for enable a recommendation, using the Generalized Extreme Value distribution. Finally, reward enforcement and behaviour policies are used to provide notifications and feedback to the student.

[1] translated: Learning by Computer Interactive System by University of Aveiro

## 1.1 Related Work

Using Bayesian inference and networks to . . . [7, 6, 1, 4, 3, 5, 2].

# 2 Mixture modelling

Imagine a student starts answering a question of difficulty $i$, while having a knowledge belief of $b_1$ for that topic. If we consider the binary set $e = \{0, 1\}$ for the two outcomes for the answer, where 0 means a correct answer and 1 an incorrect answer, we can represent the full action with a vector for the question, $\mathbf{q}_i$, a number for the belief, $b_1$, and a number for the answer, $e_1$.

$$\mathbf{q}_i = [\underbrace{0 \quad 0 \quad \dots \quad \overbrace{1}^{i^{\text{th}} \text{ position}} \quad \dots \quad 0}_{d}]^{\mathrm{T}}$$

After answering a total of $n$ questions, we can construct a question matrix, $\mathbf{Q}$, a beliefs vector, $\mathbf{b}$, and an error vector, $\mathbf{e}$, such that

$$\begin{aligned} &\mathbf{Q} = \begin{bmatrix} \mathbf{q}_1 & \mathbf{q}_2 & \mathbf{q}_3 & \dots & \mathbf{q}_n \end{bmatrix}, \quad \mathbf{Q} \in \mathcal{M}_{d \times n}\left(\{0,1\}\right) \\ &\mathbf{b} = \begin{bmatrix} b_1 & b_2 & b_3 & \dots & b_n \end{bmatrix}^{\mathrm{T}}, \quad \mathbf{b} \in \mathcal{M}_{n \times 1}\left([0,1]\right) \\ &\mathbf{e} \in \{0,1\}^n . \end{aligned}$$

The matrix $\mathbf{Q}$ is called in this paper a *XOR Matrix by columns*, meaning that $\mathbf{Q}^{\mathrm{T}} \cdot \mathbb{1}_d = \mathbb{1}_n$, where

$$\mathbb{1}_n = [\underbrace{1 \quad 1 \quad \dots \quad 1}_{n}]^{\mathrm{T}}.$$

This happens because a single question cannot have two different difficulty levels. So, the matrix $\mathbf{Q}$ provides information about the questions' difficulty and the vector $\mathbf{e}$ provides the evidence of the answers. What is the vector $\mathbf{b}$ used for? Since each entry represents the level of belief that the student has knowledge about the topic, we can use those values to estimate the probability of failing a question of a certain difficulty. For that, an initial piecewise function $\phi$ was created to map a level of belief to the corresponding probability of answering incorrectly.

$$\phi(x) = \begin{cases} 1 - p_g, & \text{x} < 0.05 \\ \frac{p_s - (1-p_g)}{0.9}(x - 0.05) + (1 - p_g), & 0.05 < \text{x} < 0.95 \\ p_s, & \text{x} > 0.95 \end{cases}$$

This mapping is mainly linear, considering a probability of just guessing the right answer, $p_g$, and the probability of a *slip*, a mistake on the calculations, $p_s$. Applying this function to each element of $\mathbf{b}$, yields the vector

$$\mathbf{b}|_{\phi} = \begin{bmatrix} \phi(b_1) & \phi(b_2) & \dots & \phi(b_n) \end{bmatrix}^{\mathrm{T}} = \boldsymbol{\phi}_{\mathbf{b}}.$$

This vector presents a rough estimation about the probability of failing each one of the questions, but how do we find the general probability? If we consider each question as a variable that follows a Bernoulli distribution, the probability $\vartheta$ of success of that distribution is somehow derived from the vector $\boldsymbol{\phi}_{\mathbf{b}}$.

Lets suppose that $\boldsymbol{\phi} = (X_1, X_2, X_3, \dots, X_n)$, where $X_i \sim \text{Beta}(\alpha, \beta)$. Using the method of moments, we can estimate values for $\hat{\alpha}$ and $\hat{\beta}$. Let $\bar{X}$ be the sample mean and $S^2$ the sample variance.

$$\begin{aligned} \hat{\alpha} &= \bar{X}\left[\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right], \\ \hat{\beta} &= (1-\bar{X})\left[\frac{\bar{X}(1-\bar{X})}{S^2} - 1\right]. \end{aligned}$$

Regarding $\boldsymbol{\phi}_{\mathbf{b}}$ as an observation of $\boldsymbol{\phi}$, we can use the distribution $\text{Beta}\left(\hat{\alpha}, \hat{\beta}\right)$ as a prior for the parameter $\vartheta$ of the Bernoulli distribution, where $P(\vartheta) = \text{Beta}\left(\vartheta \mid \hat{\alpha}, \hat{\beta}\right) = \frac{\vartheta^{\hat{\alpha}-1}(1-\vartheta)^{\hat{\beta}-1}}{B\left(\hat{\alpha},\hat{\beta}\right)}$. Then, using as evidence the vector $\mathbf{e}$, we can apply the Bayes' rule to update the parameters of the Beta distribution. Before jumping to that, we need to present some notation.

We say a matrix $M$ is *row-induced* by a vector $b$ iff there is a matrix $A$ such that $M = \operatorname{diag}(b) \cdot A$, where

$$\operatorname{diag}(b) = \begin{bmatrix} b_1 & 0 & \dots & 0 \\ 0 & b_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \dots & 0 & b_n \end{bmatrix},$$

and we write $M = A \mid_r b$. Likewise, we say that $M$ is *column-induced* by a vector $b$ iff there is a matrix $A$ such that $M = A \cdot \operatorname{diag}(b)$, and we write $M = A \mid_c b$. In this paper, we also use the symbol $\circ$ as the Hadamard product between matrices.

To calculate the estimated $\hat{\alpha}_d$ and $\hat{\beta}_d$ for each level of difficulty, we have to calculate the sample mean and variance and apply the method of moments for each difficulty level. Since $\mathbf{Q}$ is a *XOR Matrix by columns* and not by rows, we have that $\mathbf{Q}^{\mathrm{T}} \cdot \mathbb{1}_d = \mathbb{1}_n$ and $\mathbf{Q} \cdot \mathbb{1}_n = \mathbf{n}_d$, where $\mathbf{n}_d$ is a vector consisting of the number of questions per difficulty level. Following the method of moments:

$$\begin{aligned} \text{Sample mean: } & \boldsymbol{\mu}_d = (\mathbf{Q} \mid_c \boldsymbol{\phi}_{\mathbf{b}} \cdot \mathbb{1}_n) \circ \mathbf{n}_d^{\circ -1} \\ \text{Sample variance: } & \mathbf{S}_d^2 = \Big( (\mathbf{Q} \mid_c \boldsymbol{\phi}_{\mathbf{b}} - \mathbf{Q} \mid_r \boldsymbol{\mu}_d) (\mathbf{Q} \mid_c \boldsymbol{\phi}_{\mathbf{b}} - \mathbf{Q} \mid_r \boldsymbol{\mu}_d)^{\mathrm{T}} \cdot \mathbb{1}_d \Big) \circ (\mathbf{n}_d - \mathbb{1}_d)^{\circ -1} \\ \text{Estimated } \alpha\text{: } & \hat{\boldsymbol{\alpha}}_d = \boldsymbol{\mu}_d \circ \left[ \boldsymbol{\mu}_d \circ (\mathbb{1}_d - \boldsymbol{\mu}_d) \circ \mathbf{S}_d^{\circ -2} - \mathbb{1}_d \right] \\ \text{Estimated } \beta\text{: } & \hat{\boldsymbol{\beta}}_d = (\mathbb{1}_d - \boldsymbol{\mu}_d) \circ \left[ \boldsymbol{\mu}_d \circ (\mathbb{1}_d - \boldsymbol{\mu}_d) \circ \mathbf{S}_d^{\circ -2} - \mathbb{1}_d \right] \end{aligned}$$

Using these results we can define a vector that holds all the priors for the Bernoulli distributions, where the vector $\boldsymbol{\vartheta}$ describes a possible learning curve and

$$P(\boldsymbol{\vartheta}) = \mathbf{Beta}\Big(\boldsymbol{\vartheta} \mid \hat{\boldsymbol{\alpha}}_d, \hat{\boldsymbol{\beta}}_d\Big) = \begin{bmatrix} \operatorname{Beta}\Big(\vartheta_1 \mid \hat{\alpha}_1, \hat{\beta}_1\Big) \\ \operatorname{Beta}\Big(\vartheta_2 \mid \hat{\alpha}_2, \hat{\beta}_2\Big) \\ \vdots \\ \operatorname{Beta}\Big(\vartheta_d \mid \hat{\alpha}_d, \hat{\beta}_d\Big) \end{bmatrix}$$

## 2.1 General Mixture Model and Bayesian Inference

Each possible learning curve can be a component of a probabilistic model, consisting on a product of Bernoulli distributions. A learning curve $\boldsymbol{\vartheta}$ represents the performance level of the student for each difficulty. The probability of the error vector $\mathbf{e}^s$ according to the learning curve $\boldsymbol{\vartheta}$ is

$$P(\mathbf{e}^s \mid \boldsymbol{\vartheta}) = \varpi_r \left( \boldsymbol{\mathcal{B}} \left( \mathbf{Q}^{\mathrm{T}} \cdot \boldsymbol{\vartheta}, \mathbf{e}^s \right) \right) = \prod_{n=1}^{N} \mathcal{B} \left( (\mathbf{Q}^{\mathrm{T}})_n \cdot \boldsymbol{\vartheta}, \mathbf{e}_n^s \right),$$

where $\varpi_r(b) = \det(\operatorname{diag}(b)) = \prod_n b_n$. So a $K$-component mixture over learning curves is a set of learning curves $\boldsymbol{\vartheta}_1, \boldsymbol{\vartheta}_2, \dots, \boldsymbol{\vartheta}_k$ with probabilities $\rho_1, \rho_2, \dots, \rho_k$, where the probability of the error vector $\mathbf{e}^s$, according to the mixture model, is

$$\sum_{k=1}^{K} \rho_k \cdot \varpi_r \left( \boldsymbol{\mathcal{B}} \left( \mathbf{Q}^{\mathrm{T}} \cdot \boldsymbol{\vartheta}_k, \mathbf{e}^s \right) \right).$$

The Bayes' rule states that $P(\boldsymbol{\vartheta} \mid \mathbf{e}) \propto P(\mathbf{e} \mid \boldsymbol{\vartheta}) P(\boldsymbol{\vartheta})$, which means that, after collecting some evidence about the error vector $\mathbf{e}$, the values for $\alpha$ and $\beta$ on $\mathbf{Beta}\Big(\hat{\boldsymbol{\alpha}}_d, \hat{\boldsymbol{\beta}}_d\Big)$ can be updated for each level of difficulty. However, and because we are working with vectors and multiple difficulty levels, the formula for the rule has to be updated to the vector form $\mathbf{P}(\boldsymbol{\vartheta} \mid \mathbf{e}) \propto \mathbf{P}(\mathbf{e} \mid \boldsymbol{\vartheta}) \circ \mathbf{P}(\boldsymbol{\vartheta})$.

$$\mathbf{P}_d(\mathbf{e} \mid \boldsymbol{\vartheta}) = \prod_{\substack{n=1 \\ \mathbf{Q}_{d,n} \neq 0}}^{N} \mathcal{B} \left( \boldsymbol{\vartheta}_d, \mathbf{e}_n \right) = \boldsymbol{\vartheta}_d^{n|_{\mathbf{e}_n = 1}} (1 - \boldsymbol{\vartheta}_d)^{n|_{\mathbf{e}_n = 0}},$$

$$\mathbf{P}_d(\boldsymbol{\vartheta}) = \operatorname{Beta}\Big(\boldsymbol{\vartheta}_d \mid \hat{\boldsymbol{\alpha}}_d, \hat{\boldsymbol{\beta}}_d\Big) = \frac{\boldsymbol{\vartheta}_d^{\hat{\boldsymbol{\alpha}}_d - 1} (1 - \boldsymbol{\vartheta}_d)^{\hat{\boldsymbol{\beta}}_d - 1}}{B\Big(\hat{\boldsymbol{\alpha}}_d, \hat{\boldsymbol{\beta}}_d\Big)},$$

$$\mathbf{P}_d(\boldsymbol{\vartheta} \mid \mathbf{e}) \propto \mathbf{P}_d(\mathbf{e} \mid \boldsymbol{\vartheta}) \cdot \mathbf{P}_d(\boldsymbol{\vartheta}) = \frac{\boldsymbol{\vartheta}_d^{\hat{\boldsymbol{\alpha}}_d + n|_{\mathbf{e}_n=1} - 1} (1 - \boldsymbol{\vartheta}_d)^{\hat{\boldsymbol{\beta}}_d + n|_{\mathbf{e}_n=0} - 1}}{B\Big(\hat{\boldsymbol{\alpha}}_d + n|_{\mathbf{e}_n=1}, \hat{\boldsymbol{\beta}}_d + n|_{\mathbf{e}_n=0}\Big)} = \mathrm{Beta}\Big(\boldsymbol{\vartheta}_d \mid \dot{\boldsymbol{\alpha}}_d, \dot{\boldsymbol{\beta}}_d\Big) .$$

Now that we have calculated the posterior probability, we can normalize the parameters to get back a good value for $\dot{\vartheta}_d = \frac{\dot{\alpha}_d - 1}{\dot{\alpha}_d + \dot{\beta}_d - 2}$. The main issue about this method is that the real likelihood of $\vartheta$, given an error vector $\mathbf{e}^s$ of a student, also accounts for different difficulties, and the method above checks for right/wrong answers for each difficulty individually. Although, it's a good and simple way of getting a fitting learning curve after answering a series of questions, based on the beliefs of the topics regarding those questions.

## 2.2 Statistical Consistency

The goal is then to create a method such that the mixture model is statistical consistent, i.e., given enough data, it converges to the true probabilities. So, given data about the matices $\mathbf{Q}^s$ and error vectors $\mathbf{e}^s$, for each student $s$, we have to find an algorithm that guarantees that the student $s$ belongs to one and only one learning curve as $N \to +\infty$ and $K \to +\infty$. It is important to state that what we call *leraning curve* in this paper expresses a curve that isn't temporal nor measured relatively to the number of tries, but as a static value for each probability of making a mistake on a question of a given difficulty. This means that, although the question matrices and the error vectors should be temporal just to keep operations between them truthful, their orientation is meaningless to the mixture model, so perturbations on the matrices and vectors imply changing multiple 0's to 1's, or vice-versa.

**Theorem 1.** *Given a sufficient number of components, there exists a statistical consistent algorithm that ensures the mixture model converges to the ground truth as the number of difficulty matrices $\mathbf{Q}^s$ and error vectors $\mathbf{e}^s$ grows.*

*Proof.* To prove the theorem, for simplicity, let's first assume there's just one level of difficulty, i.e., $\mathbf{Q}^s = (\mathbb{1}_n)^{\mathrm{T}}$ and each Bernoulli distribution is of the form $\mathcal{B}(\vartheta_k, e_n^s)$. The likelihood $\mathcal{L}(\vartheta_k \mid \mathbf{e}^s) = \prod_n \mathcal{B}(\vartheta_k, e_n^s)$ can be maximized by calculating $\frac{\partial}{\partial \vartheta_k} \mathcal{L}(\vartheta_k \mid \mathbf{e}^s) = 0$, where $\vartheta_k \in ]0, 1[$. Because there's only one difficulty level, $\mathcal{L}(\vartheta_k \mid \mathbf{e}^s) = \vartheta_k^w (1 - \vartheta_k)^r$, where $r \geq 1$ and $w \geq 1$ mean the number of right and wrong answers, and

$$\begin{aligned}
0 &= \frac{\partial}{\partial \vartheta_k} \mathcal{L}(\vartheta_k \mid \mathbf{e}^s) \\
0 &= \vartheta_k^{w-1} (1 - \vartheta_k)^{r-1} \left(w - \vartheta_k (w + r)\right) \\
0 &= \left(w - \vartheta_k (w + r)\right) \\
\vartheta_k &= \frac{w}{w + r}.
\end{aligned}$$

This means that for an error vector $\mathbf{e}^s$, where $w = (\mathbf{e}^s)^{\mathrm{T}} \cdot \mathbb{1}_n$ and $r = n - w$, there exists an exact value for $\vartheta_k$ that maximizes the likelihood. So, as $K \to +\infty$,

$$\forall \varepsilon > 0 \ \ \exists k \leq K \in \mathbb{N} \ \ \left| \vartheta_k - \frac{w}{w + r} \right| < \varepsilon. \tag{1}$$

However, how do we know that this algorithm provides stability to the system? Regarding perturbations on the error vectors $\mathbf{e}^{s'}$, how do they influence the value of likelihood $\mathcal{L}\Big(\vartheta_k \mid \mathbf{e}^{s'}\Big)$? Considering a fixed perturbation $t \geq 1$, the values for right and wrong answers change to $w' = w \pm t$ and $r' = r \mp t$, and so the $\vartheta$ that maximizes likelihood would be $\vartheta_k^{\pm} = \frac{w \pm t}{w + r}$, and that doesn't guarantee the truth of (1) for $\mathbf{e}^s$ and $\mathbf{e}^{s'}$ simultaneously. Nonetheless, as $n \to +\infty$, $w \pm t \approx w$, and a learning curve that is "close enough" would also be fitting to that perturbation. So, for a fixed perturbation $t \neq 0$, generating the vector $\mathbf{e}^{s'}$,

$$\forall \delta > 0 \ \ \exists \varepsilon > 0 \ \ \exists n \in \mathbb{N} \ \ \left| \frac{t}{w + r} \right| < \varepsilon \Rightarrow \left| \mathcal{L}\left( \frac{w + t}{w + r} \mid \mathbf{e}^{s'} \right) - \mathcal{L}\left( \frac{w + t}{w + r} \mid \mathbf{e}^{s} \right) \right| < \delta, \tag{2}$$

which means that, joining (1) and (2),

$$\forall \delta > 0 \ \ \exists \varepsilon > 0 \ \ \exists n, k \in \mathbb{N} \ \ \left| \vartheta_k - \frac{w}{w + r} \right| < \left| \frac{t}{w + r} \right| < \varepsilon \Rightarrow \left| \mathcal{L}\Big(\vartheta_k \mid \mathbf{e}^{s'}\Big) - \mathcal{L}(\vartheta_k \mid \mathbf{e}^s) \right| < \delta, \tag{3}$$

which proves that, for one difficulty, given sufficient components and provided enough data points, each error vector will converge on a single component, i.e., the one that maximizes the likelihood, which is the same as saying, the one that minimizes the distance $\left|\vartheta_k - \frac{w}{w+r}\right|$, and, as $n \to +\infty$, error vectors with relatively small perturbations will converge to a true value for $\vartheta$. This proves the statistical consistency to a model of just one difficulty.

However, regarding $d$ difficulties, the likelihood function is of the type

$$\mathcal{L}(\boldsymbol{\vartheta} \mid \mathbf{e}^s) = \vartheta_1^{w_1}(1-\vartheta_1)^{r_1} \cdot \vartheta_2^{w_2}(1-\vartheta_2)^{r_2} \cdot \ldots \cdot \vartheta_d^{w_d}(1-\vartheta_d)^{r_d} = f_1(\vartheta_1) f_2(\vartheta_2) \ldots f_d(\vartheta_d),$$

where each $f_i \colon (0,1) \longrightarrow (0,1)$, which means that the maximum likelihood is achieved when the maximum of each $f_i$ is achieved. Following the same reasoning as before, given a fixed perturbation vector $\mathbf{t}$, (3) can be reconstructed as:

$$\forall \delta > 0 \ \ \exists \boldsymbol{\varepsilon} > \mathbb{0}_d \ \ \exists n, k \in \mathbb{N} \ \ \left|\boldsymbol{\vartheta}_k - \mathbf{w} \circ \mathbf{n}^{\circ -1}\right| < \left|\mathbf{t} \circ \mathbf{n}^{\circ -1}\right| < \boldsymbol{\varepsilon} \Rightarrow \left|\mathcal{L}\left(\boldsymbol{\vartheta}_k \mid \mathbf{e}^{s'}\right) - \mathcal{L}(\boldsymbol{\vartheta}_k \mid \mathbf{e}^s)\right| < \delta. \tag{4}$$

This proves the statistical consistency of the model for $d$ difficulties and an algorithm can be constructed, where the likelihood is calculated for every student and component and then normalized, each $\boldsymbol{\vartheta}_k$ is calculated with bayesian inference based on the beliefs of the questions or beliefs predetermined for the components and the values of $\rho_k$ are calculated based on the empirical frequency of $\mathbf{e}^s$ within the dataset. Each iteration is of the form:

$$\begin{aligned}
\mathbf{L}_{s,k} &= \rho_k \cdot \varpi_r \left(\boldsymbol{\mathcal{B}}\left(\mathbf{Q}^{\mathrm{T}} \cdot \boldsymbol{\vartheta}_k, \mathbf{e}^s\right)\right) \\
\mathbf{Z}_{s,k} &= \frac{\mathbf{L}_{s,k}}{\left(\mathbf{L} \cdot \mathbb{1}_K\right)_s} \\
\boldsymbol{\vartheta}_k &= \left(\boldsymbol{\alpha}_k - \mathbb{1}_d + \sum_s \left(\mathbf{Z}_{s,k} \cdot \mathbf{w}^s\right)\right) \circ \left(\boldsymbol{\alpha}_k + \boldsymbol{\beta}_k - \mathbb{2}_d + \sum_s \left(\mathbf{Z}_{s,k} \cdot \mathbf{n}^s\right)\right)^{\circ -1} \\
\boldsymbol{\rho} &= \frac{\mathbf{Z}^{\mathrm{T}} \cdot \mathbb{1}_S}{(\mathbb{1}_S)^{\mathrm{T}} \cdot \mathbf{Z} \cdot \mathbb{1}_K},
\end{aligned} \tag{5}$$

where $\mathbf{w}^s = (\mathbf{Q}^s \mid_c \mathbf{e}^s) \cdot \mathbb{1}_N$ and $\mathbf{n}^s = \mathbf{Q}^s \cdot \mathbb{1}_N$. □

### 2.3 Model Predictions

After implementing this algorithm and collecting enough data, we are predicting two types of learning curves: *exponential* and *logarithmic*. Both describe how the student adapts to higher levels of difficulty. The curves will be of the form:

$$\vartheta(d) = ae^{b(d-1)}, \quad 0 < a < 1, \quad 0 < b < -\frac{\ln a}{4}, \tag{6}$$

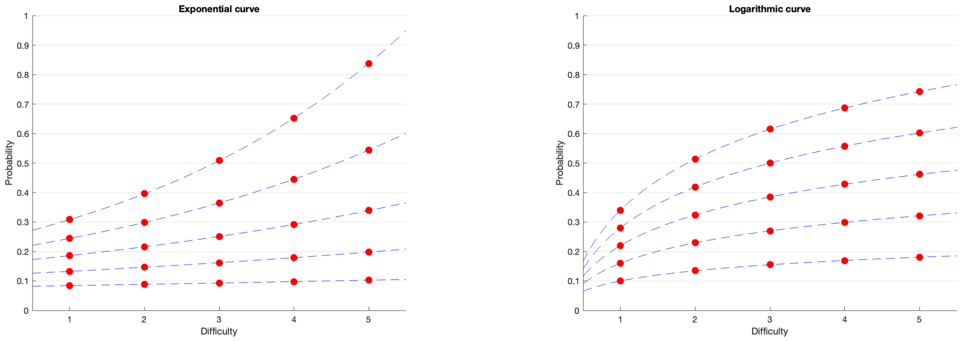
$$\vartheta(d) = a + b \ln d, \quad 0 < a < 1, \quad 0 < b < \frac{1-a}{\ln 5}, \tag{7}$$

where $a$ means the probability of answering incorrectly a question of level 1 and $b$ describes the steepness of the curve.

After computing all the learning curves, it is possible to make exponential and logarithmic regressions on those curves, to find the parameters $a$ and $b$. Then, the residuals are calculated and the regression with smaller residuals gets assigned to the learning curve. Finally, to get two variables within the same range of values, $]0,1[$, we retrieve the value $a$ and the value $b' = -\frac{4b}{\ln a}$, for (6), or $b' = \frac{b \ln 5}{1-a}$, for (7).

## 3 Tendency distribution

Imagine you want to model a tendency, consisting on one specific task $t$ at a time. Moreover, the probability of not succeeding on that task, $s$, called *slip*, influences whether or not $t$ is completed everytime. That said, the probability of succeeding on one task is $p(x) = (1-s)^x s^{1-x}$, with $x \in \{0,1\}$. If we regard each task as an independent variate, with fixed probability, then $p(\mathbf{v}) = \prod_{i=1}^n (1-s)^{v_i} s^{1-v_i}$, with $\mathbf{v} = \{0,1\}^n$. However, since we want to model tendencies, each task will have a different impact on the overall probability, proportional to their frequency of successes. This way, instead of multiplying



(a) Five possible learning curves following the exponential equation $\vartheta(d) = ae^{b(d-1)}$.

(b) Five possible learning curves following the logarithmic equation $\vartheta(d) = a + b \ln d$.

Figure 1: Predictions of the *exponential* (1a) and the *logarithmic* (1b) curves.

all probabilities individually, we sum them with predetermined weights, and normalize so all possible outcomes sum up to 1. So the formula is of the form

$$p(\mathbf{v}) = \frac{\sum_{i=1}^{n} \gamma_i (1-s)^{v_i} s^{1-v_i}}{C(n) \cdot \sum_{i=1}^{n} \gamma_i}, \tag{8}$$

with $C(n) \colon \mathbb{N} \longrightarrow \mathbb{R}$, $\forall i, j \; \gamma_i \gamma_j \geq 0$ and $\boldsymbol{\gamma} \neq \mathbb{0}_n$. Also, we force $\forall i, j \; \gamma_i \gamma_j \geq 0$ to guarantee that $p(\mathbf{v})$ only returns values between 0 and 1.

**Theorem 2.** *Regarding the equation* (8), *exists* $C(n) \colon \mathbb{N} \longrightarrow \mathbb{R}$ *such that,*

$$\boldsymbol{\gamma} \neq \mathbb{0}_n \Rightarrow \sum_{\mathbf{v} \in \{0,1\}^n} p(\mathbf{v}) = 1.$$

*Proof.* First we need to convert the equation (8) into its matrix form.

$$\begin{aligned}
\mathbf{v}_i \in \{0,1\} \quad &\Rightarrow \; (1-s)^{v_i} s^{1-v_i} = v_i(1-s) + s(1-v_i) \\
\Leftrightarrow \quad \sum_{i=1}^{n} \gamma_i (1-s)^{v_i} s^{1-v_i} &= \begin{bmatrix} v_1(1-2s)+s & v_2(1-2s)+s & \ldots & v_n(1-2s)+s \end{bmatrix} \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \vdots \\ \gamma_n \end{bmatrix} \\
&= \left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot \boldsymbol{\gamma}. \qquad (9)
\end{aligned}$$

$$C(n) \cdot \sum_{i=1}^{n} \gamma_i = C(n) \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}. \tag{10}$$

Combining (9) and (10),

$$p(\mathbf{v}) = \frac{\left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot \boldsymbol{\gamma}}{C(n) \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}}. \tag{11}$$

To prove the theorem, we have to sum all values of $p(\mathbf{v})$, where $\mathbf{v} \in \{0,1\}^n$, and find for what values of $C(n)$ the statement $\sum_{\mathbf{v} \in \{0,1\}^n} p(\mathbf{v}) = 1$. For that, lets consider the matrix $\mathbb{V}_n$ where every row is an unique vector $\mathbf{v}_i \in \{0,1\}^n$, that is, $\mathbb{V}_n \in \{0,1\}^{2^n \times n}$ such as

$$\mathbb{V}_n = \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \ldots & \mathbf{v}_{2^n} \end{bmatrix}^{\mathrm{T}}.$$

With $\boldsymbol{\gamma} \neq \mathbb{0}_n$,

$$\sum_{\mathbf{v} \in \{0,1\}^n} p(\mathbf{v}) = \frac{\sum_{\mathbf{v} \in \{0,1\}^n} \left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot \boldsymbol{\gamma}}{C(n) \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}}$$

$$
\begin{aligned}
&= \frac{\left((1-2s)\cdot \mathbb{1}^{\mathrm{T}}_{|\{0,1\}^n|}\cdot \mathbb{V}_n + s\,|\{0,1\}^n|\cdot \mathbb{1}^{\mathrm{T}}_n\right)\cdot \boldsymbol{\gamma}}{C(n)\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\\
&= \frac{\left(2^{n-1}(1-2s)\cdot \mathbb{1}^{\mathrm{T}}_n + 2^n s\cdot \mathbb{1}^{\mathrm{T}}_n\right)\cdot \boldsymbol{\gamma}}{C(n)\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\\
&= \frac{2^{n-1}\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}{C(n)\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\\
&= \frac{2^{n-1}}{C(n)}.
\end{aligned}
$$

If we set $C(n)=2^{n-1}$, then $\sum_{\mathbf{v}\in\{0,1\}^n} p(\mathbf{v})=1$ holds for any $\gamma_i \in \mathbb{R}^+_0$ such that $\boldsymbol{\gamma}\neq \mathbb{0}_n$. □

Based on the above theorem, let's start defining our tendency distribution. Let $V \sim \mathrm{Tend}\,(s,\boldsymbol{\gamma})$ be a vector-valued random variate, that follows a tendency distribution, where $s$ represents the *slip*, $n$ is the number of trials and $\boldsymbol{\gamma}$ is the weights vector, such that $\forall i,j\ \gamma_i\gamma_j \geq 0$ and $\boldsymbol{\gamma}\neq \mathbb{0}_n$. We can start by defining the probability density function (PDF):

$$
\begin{aligned}
f\colon\ \{0,1\}^n &\longrightarrow [0,1]\\
\mathbf{v} &\longmapsto \frac{\left((1-2s)\cdot \mathbf{v}^{\mathrm{T}} + s\cdot \mathbb{1}^{\mathrm{T}}_n\right)\cdot \boldsymbol{\gamma}}{2^{n-1}\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}.
\end{aligned}
\tag{12}
$$

**Theorem 3.** *Regarding the PDF* (12)*, and the random variate* $V\sim \mathrm{Tend}(s,\boldsymbol{\gamma})$*, the expected value and scalar variance are:*

$$
\mathrm{E}[V] = \frac{1}{2}\left(\frac{1-2s}{\mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\cdot \boldsymbol{\gamma} + \mathbb{1}_n\right), \tag{13}
$$

$$
\mathrm{Var}[V] = \frac{1}{2}\left(\mathbb{1}_n - \frac{1-2s}{\mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\cdot \boldsymbol{\gamma}\right)^{\mathrm{T}}\cdot \mathrm{E}[V]. \tag{14}
$$

*Proof.* To prove the expected value, we just need to use algebra to achieve the answer.

$$
\begin{aligned}
\mathrm{E}[V] &= \sum_{\mathbf{v}\in\{0,1\}^n} f(\mathbf{v})\cdot \mathbf{v}\\
&= \mathbb{V}^{\mathrm{T}}_n\cdot \mathbf{f}(\mathbb{V}_n),
\end{aligned}
$$

where $\mathbf{f}$ is the matrix representation of the PDF $f$, for all possible values of $\mathbf{v}\in\{0,1\}^n$. Since $\mathbb{V}_n$ is constructed such as all possible values $\mathbf{v}$ are distributed on the rows, we just need to change $\mathbf{v}^{\mathrm{T}}$ to $\mathbb{V}_n$ and $\mathbb{1}^{\mathrm{T}}_n$ to $\mathbb{1}_{2^n\times n}$ in $f$. This way,

$$
\mathbf{f}(\mathbb{V}_n) = \frac{\left((1-2s)\cdot \mathbb{V}_n + s\cdot \mathbb{1}_{2^n\times n}\right)\cdot \boldsymbol{\gamma}}{2^{n-1}\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}.
$$

Using this result,

$$
\begin{aligned}
\mathrm{E}[V] &= \mathbb{V}^{\mathrm{T}}_n\cdot \mathbf{f}(\mathbb{V}_n)\\
&= \frac{\mathbb{V}^{\mathrm{T}}_n\cdot\left((1-2s)\cdot \mathbb{V}_n + s\cdot \mathbb{1}_{2^n\times n}\right)\cdot \boldsymbol{\gamma}}{2^{n-1}\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\\
&= \frac{\left((1-2s)\cdot \mathbb{V}^{\mathrm{T}}_n\cdot \mathbb{V}_n + s\cdot \mathbb{V}^{\mathrm{T}}_n\cdot \mathbb{1}_{2^n\times n}\right)\cdot \boldsymbol{\gamma}}{2^{n-1}\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}.
\end{aligned}
$$

By the properties of the matrix $\mathbb{V}_n$, we can calculate $\mathbb{V}^{\mathrm{T}}_n\cdot \mathbb{V}_n = 2^{n-2}\left(I_n + \mathbb{1}_{n\times n}\right)$, where $I_n=\mathrm{diag}(\mathbb{1}_n)$ is the identity matrix of size $n$, and $\mathbb{V}^{\mathrm{T}}_n\cdot \mathbb{1}_{2^n\times n} = 2^{n-1}\cdot \mathbb{1}_{n\times n} = 2\left(2^{n-2}\cdot \mathbb{1}_{n\times n}\right)$. So,

$$
\begin{aligned}
\mathrm{E}[V] &= \frac{\left(2^{n-2}(1-2s)I_n + (1-2s)\left(2^{n-2}\cdot \mathbb{1}_{n\times n}\right) + 2s\left(2^{n-2}\cdot \mathbb{1}_{n\times n}\right)\right)\cdot \boldsymbol{\gamma}}{2^{n-1}\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}\\
&= \frac{\left((1-2s)I_n + \mathbb{1}_{n\times n}\right)\cdot \boldsymbol{\gamma}}{2\cdot \mathbb{1}^{\mathrm{T}}_n\cdot \boldsymbol{\gamma}}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{1}{2}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\cdot\boldsymbol{\gamma} + \frac{\mathbb{1}_n\cdot\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}{\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\right)\\
&= \frac{1}{2}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\cdot\boldsymbol{\gamma} + \mathbb{1}_n\right),
\end{aligned}
$$

proving (13). For the variance, we define $\boldsymbol{\mu} = \mathrm{E}[V]$ and the definition for vector-valued random variates $\mathrm{Var}[V] = \mathrm{E}\left[(V-\boldsymbol{\mu})^{\mathrm{T}}(V-\boldsymbol{\mu})\right]$ to find the result.

$$
\begin{aligned}
\mathrm{Var}[V] &= \mathrm{E}\left[(V-\boldsymbol{\mu})^{\mathrm{T}}(V-\boldsymbol{\mu})\right]\\
&= \sum_{\mathbf{v}\in\{0,1\}^n} f(\mathbf{v})\cdot(\mathbf{v}-\boldsymbol{\mu})^{\mathrm{T}}(\mathbf{v}-\boldsymbol{\mu})\\
&= \sum_{\mathbf{v}\in\{0,1\}^n} f(\mathbf{v})R(\mathbf{v})\\
&= \mathrm{Diag}(\mathbf{R}(\mathbb{V}_n))^{\mathrm{T}}\cdot\mathbf{f}(\mathbb{V}_n),
\end{aligned}
$$

where $R(\mathbf{v}) = (\mathbf{v}-\boldsymbol{\mu})^{\mathrm{T}}(\mathbf{v}-\boldsymbol{\mu})$, $\mathrm{Diag}(M) = (M\circ I_n)\cdot\mathbb{1}_n$ is a linear function that returns the diagonal of a matrix as a column vector, and $\mathbf{R}$ is the matrix representation of the function $R$ for all possible values of $\mathbf{v}\in\{0,1\}^n$. We now have to find $\mathrm{Diag}(\mathbf{R}(\mathbb{V}_n))$. Similarly to $\mathbf{f}$, if $R$ uses the unique vector as a column, in the matrix representation we replace it with $\mathbb{V}_n^{\mathrm{T}}$. To respect the matrix dimensions, we need to extend the vector $\boldsymbol{\mu}$ to the matrix $\boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}$. This way,

$$
\mathbf{R}(\mathbb{V}_n) = \left(\mathbb{V}_n^{\mathrm{T}} - \boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}\right)^{\mathrm{T}}\left(\mathbb{V}_n^{\mathrm{T}} - \boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}\right).
$$

Due to the distributive property of matrices and the linearity of the Diag function,

$$
\begin{aligned}
\mathrm{Diag}(\mathbf{R}(\mathbb{V}_n)) &= \mathrm{Diag}\Big(\mathbb{V}_n\cdot\mathbb{1}_n^{\mathrm{T}} - \left(\mathbb{V}_n\cdot\boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}\right)^{\mathrm{T}} - \mathbb{V}_n\cdot\boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}} + \boldsymbol{\mu}^2\cdot\mathbb{1}_{2^n\times 2^n}\Big).\\
\mathrm{Diag}\big(\mathbb{V}_n\cdot\mathbb{1}_n^{\mathrm{T}}\big) &= \mathbb{V}_n\cdot\mathbb{1}_n,\\
\mathrm{Diag}\Big(\left(\mathbb{V}_n\cdot\boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}\right)^{\mathrm{T}}\Big) &= \mathrm{Diag}\big(\mathbb{V}_n\cdot\boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}\big),\\
\mathrm{Diag}\big(\mathbb{V}_n\cdot\boldsymbol{\mu}\cdot\mathbb{1}_{2^n}^{\mathrm{T}}\big) &= \mathbb{V}_n\cdot\boldsymbol{\mu},\\
\mathrm{Diag}(\mathbb{1}_{2^n\times 2^n}) &= \mathbb{1}_{2^n},
\end{aligned}
$$

where $\boldsymbol{\mu}^2 = \boldsymbol{\mu}^{\mathrm{T}}\cdot\boldsymbol{\mu} = \|\boldsymbol{\mu}\|^2$, which means,

$$
\begin{aligned}
\mathrm{Diag}(\mathbf{R}(\mathbb{V}_n)) &= \mathbb{V}_n\cdot\mathbb{1}_n - 2\cdot\mathbb{V}_n\cdot\boldsymbol{\mu} + \boldsymbol{\mu}^2\cdot\mathbb{1}_{2^n} && (15)\\
&= \mathbb{V}_n\cdot(\mathbb{1}_n - 2\cdot\boldsymbol{\mu}) + \boldsymbol{\mu}^2\cdot\mathbb{1}_{2^n}. && (16)
\end{aligned}
$$

Before finishing the proof, we present some auxiliary results, important to some steps of the calculations.

$$
\begin{aligned}
(\mathbb{1}_n - 2\cdot\boldsymbol{\mu}) &= -\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\cdot\boldsymbol{\gamma},\\
\boldsymbol{\mu}^2 &= \frac{1}{4}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\cdot\boldsymbol{\gamma} + \mathbb{1}_n\right)^2 = \frac{1}{4}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\cdot\boldsymbol{\gamma}\right)^2 + \frac{1-2s}{2} + \frac{1}{4}\left(\mathbb{1}_n\right)^2,\\
\mathbb{1}_{2^n}^{\mathrm{T}}\cdot\mathbb{1}_{2^n\times n} &= 2^n\cdot\mathbb{1}_n^{\mathrm{T}} = 2\left(2^{n-1}\cdot\mathbb{1}_n^{\mathrm{T}}\right),\\
\mathbb{1}_{2^n}^{\mathrm{T}}\cdot\mathbb{V}_n &= 2^{n-1}\cdot\mathbb{1}_n^{\mathrm{T}}.
\end{aligned}
$$

We can now find the value for the variance.

$$
\begin{aligned}
\mathrm{Var}[V] &= \mathrm{Diag}(\mathbf{R}(\mathbb{V}_n))^{\mathrm{T}}\cdot\mathbf{f}(\mathbb{V}_n)\\
&= \frac{\left(\mathbb{V}_n\cdot(\mathbb{1}_n - 2\cdot\boldsymbol{\mu}) + \boldsymbol{\mu}^2\cdot\mathbb{1}_{2^n}\right)^{\mathrm{T}}\cdot((1-2s)\cdot\mathbb{V}_n + s\cdot\mathbb{1}_{2^n\times n})\cdot\boldsymbol{\gamma}}{2^{n-1}\cdot\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}\\
&= \frac{\left[(1-2s)\left(\mathbb{1}_n - 2\boldsymbol{\mu}\right)^{\mathrm{T}}\mathbb{V}_n^{\mathrm{T}}\mathbb{V}_n + s\left(\mathbb{1}_n - 2\boldsymbol{\mu}\right)^{\mathrm{T}}\mathbb{V}_n^{\mathrm{T}}\mathbb{1}_{2^n\times n} + (1-2s)\boldsymbol{\mu}^2\mathbb{1}_{2^n}^{\mathrm{T}}\mathbb{V}_n + s\boldsymbol{\mu}^2\mathbb{1}_{2^n}^{\mathrm{T}}\mathbb{1}_{2^n\times n}\right]\cdot\boldsymbol{\gamma}}{2^{n-1}\cdot\mathbb{1}_n^{\mathrm{T}}\cdot\boldsymbol{\gamma}}
\end{aligned}
$$

$$
\begin{aligned}
&= \frac{\left[2^{n-2}(1-2s)\left(\mathbb{1}_n - 2\boldsymbol{\mu}\right)^{\mathrm{T}} \cdot I_n + \left(\mathbb{1}_n - 2\boldsymbol{\mu}\right)^{\mathrm{T}} \left(2^{n-2} \cdot \mathbb{1}_{n\times n}\right) + 2^{n-1}\boldsymbol{\mu}^2 \mathbb{1}_n^{\mathrm{T}}\right] \cdot \boldsymbol{\gamma}}{2^{n-1} \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \\
&= \frac{\left[(1-2s)\left(\mathbb{1}_n - 2\boldsymbol{\mu}\right)^{\mathrm{T}} \cdot I_n + \left(\mathbb{1}_n - 2\boldsymbol{\mu}\right)^{\mathrm{T}} \cdot \mathbb{1}_{n\times n} + 2\boldsymbol{\mu}^2 \mathbb{1}_n^{\mathrm{T}}\right] \cdot \boldsymbol{\gamma}}{2 \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \\
&= \frac{-\frac{(1-2s)^2}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}^{\mathrm{T}}\boldsymbol{\gamma} - (1-2s) \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma} + 2\boldsymbol{\mu}^2 \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}}{2 \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \\
&= -\frac{1}{2}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}\right)^2 + \frac{\left(-(1-2s) + 2\boldsymbol{\mu}^2\right) \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}}{2 \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \\
&= -\frac{1}{2}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}\right)^2 + \frac{-(1-2s) + \frac{1}{2}\left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}\right)^2 + (1-2s) + \frac{1}{2} \cdot \left(\mathbb{1}_n\right)^2}{2} \\
&= \frac{1}{4}\left(\left(\mathbb{1}_n\right)^2 - \left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}\right)^2\right) \\
&= \frac{1}{2}\left(\mathbb{1}_n - \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}\right)^{\mathrm{T}} \cdot \mathrm{E}[V],
\end{aligned}
$$

proving (14). □

It's easy to notice that both $\mathrm{E}[V]$ and $\mathrm{Var}[V]$ have a common expression, that we'll call $\boldsymbol{\rho}$, and

$$
\begin{aligned}
\boldsymbol{\rho} &= \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}, \\
\mathrm{E}[V] &= \frac{1}{2}\left(\mathbb{1}_n + \boldsymbol{\rho}\right), \\
\mathrm{Var}[V] &= \frac{1}{4}\left(\mathbb{1}_n - \boldsymbol{\rho}\right)^{\mathrm{T}}\left(\mathbb{1}_n + \boldsymbol{\rho}\right) = \frac{1}{4}\left(n - \boldsymbol{\rho}^2\right),
\end{aligned}
$$

and $s \lesseqgtr \frac{1}{2} \Rightarrow \boldsymbol{\rho} \gtreqless \mathbb{0}_n$. Before looking into the statistical inference of this model, we need to state the following lemma and theorem.

**Lemma 1.** *Let* $T(\boldsymbol{\gamma}) = \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}}$*. So* $\boldsymbol{\rho} = T(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma}$*, and*

$$
s \neq \frac{1}{2} \Rightarrow \boldsymbol{\rho} = T(\boldsymbol{\rho}) \cdot \boldsymbol{\rho}.
$$

*Proof.*

$$
\begin{aligned}
& & \boldsymbol{\rho} &= T(\boldsymbol{\rho}) \cdot \boldsymbol{\rho} \\
&\Leftrightarrow & 1 &= T(\boldsymbol{\rho}) \\
&\Leftrightarrow & 1 &= \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\rho}} \\
&\Leftrightarrow & 1 &= \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \left(T(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma}\right)} \\
&\Leftrightarrow & 1 &= \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \left(\frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \cdot \boldsymbol{\gamma}\right)} \\
&\Leftrightarrow & 1 &= \frac{1-2s}{1-2s} \\
&\Leftrightarrow & 1 &= 1,
\end{aligned}
$$

that holds if $1 - 2s \neq 0 \Leftrightarrow s \neq \frac{1}{2}$. □

The proof of this lemma shows that $\boldsymbol{\rho}$ is a fixed-point just after one iteration, using $\boldsymbol{\gamma}$ as starting weights vector.

**Theorem 4.** *Let $\boldsymbol{\rho} = T(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma}$ and $V \sim \mathrm{Tend}(s, \boldsymbol{\gamma})$ a random variate. So*

$$\hat{f}(\mathbf{v}) = \frac{\left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot \boldsymbol{\rho}}{2^{n-1} \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\rho}}$$

*is also a PDF for $V$ and $\hat{f} \cong f$.*

*Proof.* By the same proof of the theorem 2, we can conclude that $\sum_{\mathbf{v} \in \{0,1\}^n} \hat{f}(\mathbf{v}) = 1$. Now let's divide the proof in two cases. First, if $s = \frac{1}{2}$, then

$$\hat{f}(\mathbf{v}) = \frac{1}{2^n},$$

which means the PDF is independent of both the vectors $\mathbf{v}$ and $\boldsymbol{\rho}$, and holds the characteristics of the PDF.

If $s \neq \frac{1}{2}$, by the proof ot the theorem 3, we achieve

$$\begin{aligned}
\hat{\boldsymbol{\rho}} &= \frac{1-2s}{\mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\rho}} \cdot \boldsymbol{\rho} = T(\boldsymbol{\rho}) \cdot \boldsymbol{\rho}, \\
\widehat{\mathrm{E}}[V] &= \frac{1}{2}\left(\mathbb{1}_n + \hat{\boldsymbol{\rho}}\right), \\
\widehat{\mathrm{Var}}[V] &= \frac{1}{4}\left(\mathbb{1}_n - \hat{\boldsymbol{\rho}}'\right)^{\mathrm{T}}\left(\mathbb{1}_n + \hat{\boldsymbol{\rho}}'\right).
\end{aligned}$$

However, by the lemma 1, $\boldsymbol{\rho} = T(\boldsymbol{\rho}) \cdot \boldsymbol{\rho} = \hat{\boldsymbol{\rho}}$, which means $\widehat{\mathrm{E}}[V] = \mathrm{E}[V]$ and $\widehat{\mathrm{Var}}[V] = \mathrm{Var}[V]$, thus proving that $\hat{f}$ is also a PDF for $V$, that retains the same expected value and scalar variance.

Furthermore, $\hat{f} \cong f$ because

$$\begin{aligned}
\hat{f}(\mathbf{v}) &= \frac{\left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot \boldsymbol{\rho}}{2^{n-1} \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\rho}} \\
&= \frac{\left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot (T(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma})}{2^{n-1} \cdot \mathbb{1}_n^{\mathrm{T}} \cdot (T(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma})} \\
&= \frac{\left((1-2s) \cdot \mathbf{v}^{\mathrm{T}} + s \cdot \mathbb{1}_n^{\mathrm{T}}\right) \cdot \boldsymbol{\gamma}}{2^{n-1} \cdot \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\gamma}} \\
&= f(\mathbf{v}),
\end{aligned}$$

when $s \neq \frac{1}{2}$ and,

$$\hat{f}(\mathbf{v}) = \frac{1}{2^n} = f(\mathbf{v}),$$

when $s = \frac{1}{2}$. □

## 3.1 Statistical Model and Inference

By the properties of the tendency distribution's PDF, the vector $\mathbf{v} = \mathbb{1}_n$ corresponds to either the greastest probability or the lowest, if $s < \frac{1}{2}$ or $s > \frac{1}{2}$, respectively. But how is it possible that the vector where every outcome is successful is either the most or the least probable? It makes sense in the same way a chain of Bernoulli events also give the greatest or lowest probability when all tasks are successful, based on the value of its parameter. Moreover, in a tendency distribution, the goal is to get the most successes possible, and the weights only state the importance of a specific task to the overall probability; a task that is usually failed will have a lower weight and thus succeeding or failing doesn't affect much the probability of that vector amongst the data.

To look into the statistical model, we first introduce the function $\Gamma(\boldsymbol{\gamma}) = T(\boldsymbol{\gamma}) \cdot \boldsymbol{\gamma}$, noting that $\boldsymbol{\rho} = \Gamma(\boldsymbol{\rho})$. This function has degree 0 of homogeneity, because $\Gamma(t \cdot \boldsymbol{\gamma}) = \Gamma(\boldsymbol{\gamma}) = \boldsymbol{\rho}$.

Suppose $V = (V_1, V_2, \ldots, V_k)$ are independent and identically distributed (i.i.d.) random variates, such that $V_i \sim \mathrm{Tend}(s, \boldsymbol{\gamma})$, with known *slip* and unknown weights. How can one estimate the value of the weights parameter? Following the empirical frequency of successes of each task on all samples, we can set the estimator

$$\widehat{\boldsymbol{\rho}}(V) = \Gamma(V \cdot \mathbb{1}_k). \tag{17}$$

Now we calculate how biased is this estimator, and the mean difference between the coordinates of the estimator and the true weights. Note that, for approximation purposes, the denominator $\mathbb{1}_n^{\mathrm{T}} \cdot V \cdot \mathbb{1}_k$ loses sense on the calculations, so we replace it for the constant $\alpha$, that accounts for the total number of successful tasks in all observations of $V$.

$$\begin{aligned}
\mathrm{E}\left[\widehat{\boldsymbol{\rho}}(V)\right] &= \mathrm{E}\left[\mathit{\Gamma}(V \cdot \mathbb{1}_k)\right] \\
&= \mathrm{E}\left[\frac{1-2s}{\alpha}\sum_{i=1}^{k} V_i\right] \\
&= \frac{k(1-2s)}{\alpha}\mathrm{E}\left[V\right] \\
&= \frac{k(1-2s)}{2\alpha}(\mathbb{1}_n + \boldsymbol{\rho}),
\end{aligned}$$

making the $\mathrm{Bias}(\widehat{\boldsymbol{\rho}}) = \frac{k(1-2s)}{2\alpha}(\mathbb{1}_n + \boldsymbol{\rho}) - \boldsymbol{\rho}$. Now we calculate the mean of each weight of the bias.

$$\begin{aligned}
\mathrm{MSD}\left(\widehat{\boldsymbol{\rho}}(V)\right) &= \frac{1}{n}\mathbb{1}_n^{\mathrm{T}} \cdot \left(\frac{k(1-2s)}{2\alpha}(\mathbb{1}_n + \boldsymbol{\rho}) - \boldsymbol{\rho}\right) \\
&= \frac{1}{n}\left[\frac{k(1-2s)}{2\alpha}\left(\mathbb{1}_n^{\mathrm{T}} \cdot \mathbb{1}_n + \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\rho}\right) - \mathbb{1}_n^{\mathrm{T}} \cdot \boldsymbol{\rho}\right] \\
&= \frac{1}{n}\left[\frac{k(1-2s)}{2\alpha}\left(n + (1-2s)\right) - (1-2s)\right] \\
&= \frac{1-2s}{n}\left[\frac{k}{2\alpha}\left(n + (1-2s)\right) - 1\right].
\end{aligned}$$

This result shows that with greater values of $n$, the closer to 0 is $\mathrm{MSD}\left(\widehat{\boldsymbol{\rho}}(V)\right)$, because (and supposing the observations of $V$ have little to none $\mathbb{0}_n$ vectors) $\alpha$ is almost everytime greater than $k$, and $\alpha \leq nk$. Similarly, as $\alpha \to nk$, $\mathrm{MSD}\left(\widehat{\boldsymbol{\rho}}(V)\right) \to \frac{(1-2s)(1-2s-n)}{2n^2}$, that approaches 0 when $n \to +\infty$. So this makes our estimator (17) a good enought approximation of the ideal weigths vector for the tendency distributions.

Knowing now the weigths vector, there is a simple way to apply inference to the parameter. Let $V^1 = (V_1, \ldots, V_i)$, $V^2 = (V_{i+1}, \ldots, V_k)$ and $V = \left(V^1, V^2\right)$ be i.d.d. random variates, where $V_j^1 \sim \mathrm{Tend}\left(s, \boldsymbol{\rho}\right)$, with known *slip* and weights, $V_j^2 \sim \mathrm{Tend}\left(s, \widetilde{\boldsymbol{\rho}}\right)$, with known *slip* and unknown weights, and $V_j \sim \mathrm{Tend}\left(s, \widehat{\boldsymbol{\rho}}\right)$, with known *slip* and unknown weights. Regarding the previous estimator,

$$\begin{aligned}
\widehat{\boldsymbol{\rho}}(V) &= \mathit{\Gamma}(V \cdot \mathbb{1}_k) \\
&= \mathit{\Gamma}((V^1 + V^2) \cdot \mathbb{1}_k) \\
&= \frac{(1-2s) \cdot (V^1 + V^2) \cdot \mathbb{1}_k}{\mathbb{1}_n^{\mathrm{T}} \cdot (V^1 + V^2) \cdot \mathbb{1}_k} \\
&= (1-2s)\frac{V^1 \cdot \mathbb{1}_k + V^2 \cdot \mathbb{1}_k}{\mathbb{1}_n^{\mathrm{T}} \cdot V^1 \cdot \mathbb{1}_k + \mathbb{1}_n^{\mathrm{T}} \cdot V^2 \cdot \mathbb{1}_k} \\
&= (1-2s)\left(\frac{V^1 \cdot \mathbb{1}_k}{\mathbb{1}_n^{\mathrm{T}} \cdot V^1 \cdot \mathbb{1}_k} \oplus \frac{V^2 \cdot \mathbb{1}_k}{\mathbb{1}_n^{\mathrm{T}} \cdot V^2 \cdot \mathbb{1}_k}\right) \\
&= \mathit{\Gamma}(V^1 \cdot \mathbb{1}_k) \oplus \mathit{\Gamma}(V^2 \cdot \mathbb{1}_k) \\
&= \boldsymbol{\rho}(V^1) \oplus \widetilde{\boldsymbol{\rho}}(V^2),
\end{aligned}$$

where the operation $\oplus$ is the Farey sum, and $s \in [0,1] \cap \mathbb{Q} \setminus \left\{\frac{1}{2}\right\}$. This means that, having the weights vector for $V^1$ and estimating the weights vector for $V^2$, we can apply inference to update the weights for $V$.

## 3.2 Model Predictions

The fact that a tendency distribution has a fixed size $n$ for each sample can be used as an advantage to model structures of fixed size. For this web application, the structure consists of the questions of a given topic, and each sample saves the first attempt of a student on all the questions, where 1 is a wrong answer and 0 otherwise.

# 4 Routine Distribution

Another way of modelling tendencies is with the creation of routines with greater periods, where the vector $\mathbf{v}$ that best fits that routine will have a greater probability. Following a similar weights approach, instead of multiplying the weights to each probability, the weight determines the probability and they are all multiplied together. So the formula is of the form

$$p(\mathbf{v}) = \frac{\prod_{i=1}^{n} \gamma_i^{v_i}}{C(\gamma)}, \tag{18}$$

with $C\colon (\mathbb{R}^+)^n \longrightarrow \mathbb{R}^+$ and $\gamma_i \in \mathbb{R}^+$. We also force $p\colon \{0,1\}^n \longrightarrow [0,1]$.

**Theorem 5.** *Regarding the equation* (18)*, exists $C\colon (\mathbb{R}^+)^n \longrightarrow \mathbb{R}^+$ such that, for all $\gamma_i \in \mathbb{R}^+$,*

$$\sum_{\mathbf{v}\in\{0,1\}^n} p(\mathbf{v}) = 1.$$

*Proof.* Since $C(\gamma)$ does not depend on $\mathbf{v}$,

$$\begin{aligned}\sum_{\mathbf{v}\in\{0,1\}^n} p(\mathbf{v}) &= \sum_{\mathbf{v}\in\{0,1\}^n} \frac{\prod_{i=1}^{n} \gamma_i^{v_i}}{C(\gamma)} \\ &= \frac{\sum_{\mathbf{v}\in\{0,1\}^n} \prod_{i=1}^{n} \gamma_i^{\mathbf{v}_i}}{C(\gamma)}.\end{aligned}$$

Due to the binary properties of the vector $\mathbf{v}$, and since we iterate over all possible values of $\mathbf{v} \in \{0,1\}^n$, we can rewrite

$$\sum_{\mathbf{v}\in\{0,1\}^n} \prod_{i=1}^{n} \gamma_i^{v_i} = \prod_{i=1}^{n} (1+\gamma_i).$$

Replacing this result,

$$\sum_{\mathbf{v}\in\{0,1\}^n} p(\mathbf{v}) = \frac{\prod_{i=1}^{n}(1+\gamma_i)}{C(\gamma)}.$$

If we set $C(\gamma) = \prod_{i=1}^{n}(1+\gamma_i)$, then $\sum_{\mathbf{v}\in\{0,1\}^n} p(\mathbf{v}) = 1$, for all $\gamma_i \in \mathbb{R}^+$. □

So let's define our routine distribution. Let $V \sim \operatorname{Rout}(\boldsymbol{\gamma})$ be a vector-valued random variate, that follows a routine distribution, where $n$ is the number of trials and $\boldsymbol{\gamma} \in (\mathbb{R}^+)^n$ is the weights vector. We can start by defining the probability density function (PDF):

$$\begin{aligned} f\colon \{0,1\}^n &\longrightarrow [0,1] \\ \mathbf{v} &\longmapsto \prod_{i=1}^{n} \frac{\gamma_i^{v_i}}{1+\gamma_i}. \end{aligned} \tag{19}$$

Before converting the PDF to its matrix form, we need to revisit the definition of the function $\varpi$. Above, we have described $\varpi_r(b) = \det(\operatorname{diag}(b))$, but we need to extend this function to accept also matrices. So we will call this method the hadamard matrix compression (by rows or columns), where given the matrices $M$ and $N$ and such that,

$$\begin{aligned} M &= \begin{bmatrix} \mathbf{v}_1 & \mathbf{v}_2 & \dots & \mathbf{v}_n \end{bmatrix}, \\ N &= \begin{bmatrix} \mathbf{w}_1 & \mathbf{w}_2 & \dots & \mathbf{w}_n \end{bmatrix}^{\mathrm{T}}, \end{aligned}$$

the functions $\varpi_c$ and $\varpi_r$ are defined as

$$\begin{aligned} \varpi_c(M) &= \mathbf{v}_1 \circ \mathbf{v}_2 \circ \cdots \circ \mathbf{v}_n, \\ \varpi_r(N) &= \mathbf{w}_1^{\mathrm{T}} \circ \mathbf{w}_2^{\mathrm{T}} \circ \cdots \circ \mathbf{w}_n^{\mathrm{T}}, \\ \varpi_c(M) &= \varpi_r(M^{\mathrm{T}})^{\mathrm{T}}. \end{aligned}$$

Now, after some simple calculations, it is possible to define the PDF in its matrix form:

$$f(\mathbf{v}) = \varpi_r\left( (\mathbf{v} \circ (\boldsymbol{\gamma} - \mathbb{1}_n) + \mathbb{1}_n) \circ (\mathbb{1}_n + \boldsymbol{\gamma})^{\circ -1} \right). \tag{20}$$

**Theorem 6.** *Regarding the PDF* (20)*, and the random variate $V$, the expected value and scalar variance are:*

$$\mathrm{E}[V] = \boldsymbol{\gamma} \circ (\mathbb{1}_n + \boldsymbol{\gamma})^{\circ -1}, \tag{21}$$

$$\mathrm{Var}[V] = \sum_{i=1}^{n} \frac{\gamma_i}{(1+\gamma_i)^2}. \tag{22}$$

*Proof.* Note first that we will use the same formulae for the $\mathrm{E}[V]$, and $\mathrm{Var}[V]$,

$$\begin{aligned}\mathrm{E}[V] &= \sum_{\mathbf{v}\in\{0,1\}^n} f(\mathbf{v})\cdot\mathbf{v}\\ &= \mathbb{V}_n^{\mathrm{T}}\cdot\mathbf{f}(\mathbb{V}_n),\\ \mathrm{Var}[V] &= \mathrm{E}\left[(V-\boldsymbol{\mu})^{\mathrm{T}}(V-\boldsymbol{\mu})\right]\\ &= \mathrm{Diag}(\mathbf{R}(\mathbb{V}_n))^{\mathrm{T}}\cdot\mathbf{f}(\mathbb{V}_n),\end{aligned}$$

where $\boldsymbol{\mu} = \mathrm{E}[V]$. From (20), we can also find the vector $\mathbf{f}(\mathbb{V}_n)$, in the same way it was done on the first tendency distribution, extending the vector $\mathbf{v}$ to the matrix $\mathbb{V}_n$ and the remaining vectors to matrices of concordant sizes.

$$\mathbf{f}(\mathbb{V}_n) = \varpi_c\left(\left(\mathbb{V}_n \circ \left(\mathbb{1}_{2^n}\cdot\boldsymbol{\gamma}^{\mathrm{T}} - \mathbb{1}_{2^n\times n}\right) + \mathbb{1}_{2^n\times n}\right)\circ\left(\mathbb{1}_{2^n\times n} + \mathbb{1}_{2^n}\cdot\boldsymbol{\gamma}^{\mathrm{T}}\right)^{\circ -1}\right).$$

We know the sum of all the values of $\mathbf{f}(\mathbb{V}_n)$ equals 1 and, since all values have the coefficient $\frac{1}{\prod_{i=1}^n(1+\gamma_i)}$ and each entry corresponds to a unique vector $\mathbf{v}$, then we know each entry of $\mathbf{f}$ is one of the terms of the product $\prod_{i=1}^n(1+\gamma_i)$; in fact, the term $j$ will be

$$\prod_{\substack{i=1\\ \mathbf{v}_{j\,i}=1}}^{n}\frac{\gamma_i}{1+\gamma_i}\prod_{\substack{i=1\\ \mathbf{v}_{j\,i}=0}}^{n}\frac{1}{1+\gamma_i}.$$

So, starting with $n=1$, it's trivial that

$$\mathrm{E}[V_1] = \frac{\gamma_1}{1+\gamma_1} = \boldsymbol{\gamma}\circ(\mathbb{1}_1+\boldsymbol{\gamma})^{\circ -1}.$$

Now, for $n \geq 2$, if we regard the vector $\mathbf{f}(\mathbb{V}_n)$, apart from the coefficients, we know each entry is one unique term of the product

$$\prod_{i=1}^{n}(1+\gamma_i) = 1\cdot\prod_{\substack{i=1\\ i\neq j}}^{n}(1+\gamma_i) + \gamma_j\cdot\prod_{\substack{i=1\\ i\neq j}}^{n}(1+\gamma_i),$$

with $1 \leq j \leq n$. So each term either belongs to 1 times the product of the remaining terms or to $\gamma_j$ times that product. By the properties of $\mathbb{V}_n$, each row $j$ of $\mathbb{V}_n^{\mathrm{T}}$, when multiplied by $\mathbf{f}(\mathbb{V}_n)$, will only contain the terms that belong to $\gamma_j\cdot\prod_{\substack{i=1\\ i\neq j}}^{n}(1+\gamma_i)$. If we multiply by the coefficient,

$$\frac{\gamma_j\cdot\prod_{\substack{i=1\\ i\neq j}}^{n}(1+\gamma_i)}{\prod_{i=1}^{n}(1+\gamma_i)} = \frac{\gamma_j}{1+\gamma_j}.$$

So each entry $j$ of $\mathbb{V}_n^{\mathrm{T}}\cdot\mathbf{f}(\mathbb{V}_n)$ is equal to $\frac{\gamma_j}{1+\gamma_j}$ and $\mathrm{E}[V] = \boldsymbol{\gamma}\circ(\mathbb{1}_n+\boldsymbol{\gamma})^{\circ -1}$, proving (21).

To prove the variance result, first we need to take a look to the vector $\mathrm{Diag}(\mathbf{R}(\mathbb{V}_n))$. With some calculations, it's possible to note that, for each row $j$ of $\mathbb{V}_n$, the $j$-th row of $\mathrm{Diag}(\mathbf{R}(\mathbb{V}_n))$ will be of the form

$$\sum_{\substack{i=1\\ \mathbf{v}_{j\,i}=0}}^{n}\left(\frac{\gamma_i}{1+\gamma_i}\right)^2 + \sum_{\substack{i=1\\ \mathbf{v}_{j\,i}=1}}^{n}\left(\frac{1}{1+\gamma_i}\right)^2.$$

This way, and again because of the properties of $\mathbb{V}_n$, the variance scalar value is

$$\mathrm{Var}[V] = \sum_{j=1}^{2^n}\left[\left(\sum_{\substack{i=1\\ \mathbf{v}_{j\,i}=0}}^{n}\left(\frac{\gamma_i}{1+\gamma_i}\right)^2 + \sum_{\substack{i=1\\ \mathbf{v}_{j\,i}=1}}^{n}\left(\frac{1}{1+\gamma_i}\right)^2\right)\left(\prod_{\substack{i=1\\ \mathbf{v}_{j\,i}=1}}^{n}\frac{\gamma_i}{1+\gamma_i}\prod_{\substack{i=1\\ \mathbf{v}_{j\,i}=0}}^{n}\frac{1}{1+\gamma_i}\right)\right]$$

$$
\begin{aligned}
&= \sum_{i=1}^{n} \left[ \sum_{\substack{j=1 \\ \mathbf{v}_{j\,i}=0}}^{2^n} \left( \frac{\gamma_i}{1+\gamma_i} \right)^2 \prod_{\substack{k=1 \\ \mathbf{v}_{j\,k}=1}}^{n} \frac{\gamma_k}{1+\gamma_k} \prod_{\substack{k=1 \\ \mathbf{v}_{j\,k}=0}}^{n} \frac{1}{1+\gamma_k} + \sum_{\substack{j=1 \\ \mathbf{v}_{j\,i}=1}}^{2^n} \left( \frac{1}{1+\gamma_i} \right)^2 \prod_{\substack{k=1 \\ \mathbf{v}_{j\,k}=1}}^{n} \frac{\gamma_k}{1+\gamma_k} \prod_{\substack{k=1 \\ \mathbf{v}_{j\,k}=0}}^{n} \frac{1}{1+\gamma_k} \right] \\
&= \sum_{i=1}^{n} \left[ \sum_{\substack{j=1 \\ \mathbf{v}_{j\,i}=0}}^{2^n} \frac{\gamma_i^2}{(1+\gamma_i)^3} \prod_{\substack{k=1 \\ k\neq i \\ \mathbf{v}_{j\,k}=1}}^{n} \gamma_k \prod_{\substack{k=1 \\ k\neq i}}^{n} \frac{1}{1+\gamma_k} + \sum_{\substack{j=1 \\ \mathbf{v}_{j\,i}=1}}^{2^n} \frac{\gamma_i}{(1+\gamma_i)^3} \prod_{\substack{k=1 \\ k\neq i \\ \mathbf{v}_{j\,k}=1}}^{n} \gamma_k \prod_{\substack{k=1 \\ k\neq i}}^{n} \frac{1}{1+\gamma_k} \right] \\
&= \sum_{i=1}^{n} \left[ \frac{\gamma_i^2}{(1+\gamma_i)^3} \cancelto{1}{\sum_{\substack{j=1 \\ \mathbf{v}_{j\,i}=0}}^{2^n} \prod_{\substack{k=1 \\ k\neq i \\ \mathbf{v}_{j\,k}=1}}^{n} \gamma_k \prod_{\substack{k=1 \\ k\neq i}}^{n} \frac{1}{1+\gamma_k}} + \frac{\gamma_i}{(1+\gamma_i)^3} \cancelto{1}{\sum_{\substack{j=1 \\ \mathbf{v}_{j\,i}=1}}^{2^n} \prod_{\substack{k=1 \\ k\neq i \\ \mathbf{v}_{j\,k}=1}}^{n} \gamma_k \prod_{\substack{k=1 \\ k\neq i}}^{n} \frac{1}{1+\gamma_k}} \right] \\
&= \sum_{i=1}^{n} \left( \frac{\gamma_i^2}{(1+\gamma_i)^3} + \frac{\gamma_i}{(1+\gamma_i)^3} \right) \\
&= \sum_{i=1}^{n} \frac{\gamma_i}{(1+\gamma_i)^2},
\end{aligned}
$$

proving (22). □

This time, the common expression $\boldsymbol{\rho}$ is

$$
\begin{aligned}
\boldsymbol{\rho} &= (\mathbb{1}_n + \boldsymbol{\gamma})^{\circ -1}, \\
\mathrm{E}[V] &= \boldsymbol{\gamma} \circ \boldsymbol{\rho} = \mathbb{1}_n - \boldsymbol{\rho}, \\
\mathrm{Var}[V] &= \mathbb{1}_n^{\mathrm{T}} \cdot \left( \boldsymbol{\gamma} \circ \boldsymbol{\rho}^{\circ 2} \right).
\end{aligned}
$$

This time, iterating $\boldsymbol{\rho}_{k+1} = (\mathbb{1}_n + \boldsymbol{\rho}_k)^{\circ -1}$ does not converge to a useful value; in fact, it can be proven that it converges to $\phi^{-1}$, where $\phi$ is the golden ratio, for any $\boldsymbol{\gamma} \in (\mathbb{R}^+)^n$. However, we can rewrite $f(\mathbf{v})$ in terms of $\boldsymbol{\rho}$,

$$
\begin{aligned}
f(\mathbf{v}) &= \prod_{i=1}^{n} \frac{\gamma_i^{v_i}}{1+\gamma_i} \\
&= \prod_{i=1}^{n} \frac{1^{1-v_i} \gamma_i^{v_i}}{(1+\gamma_i)^{1-v_i} (1+\gamma_i)^{v_i}} \\
&= \prod_{i=1}^{n} \frac{1^{1-v_i}}{(1+\gamma_i)^{1-v_i}} \frac{\gamma_i^{v_i}}{(1+\gamma_i)^{v_i}} \\
&= \prod_{i=1}^{n} \rho_i^{1-v_i} (1-\rho_i)^{v_i},
\end{aligned}
$$

which is a product of Bernoulli distributions $B_i \sim \mathcal{B}(1-\rho_i)$.

## 4.1 Statistical Model and Inference

If the PDF is equivalent to a product of Bernoulli distributions, why not defining the routine distribution that way? Although it is the same, the interpretation is rather different; in Bernoulli distributions we care about the probability of success on each task but with weights we are just setting a relative importance to them. In a routine distribution, the goal isn't getting the most successes possible but managing the successes and failures accordingly to the scheduled routine. That said, suppose $V = (V_1, V_2, \ldots, V_k)$ are i.i.d. random variates, such that $V_i \sim \mathrm{Rout}(\boldsymbol{\gamma})$, with unknown weights. Maximum likelihood estimation can be used to estimate the value of $\boldsymbol{\gamma}$, by finding the root of the derivative of the log-likelihood. Let $\mathbf{V} = (\mathbf{v}_1, \mathbf{v}_2, \ldots, \mathbf{v}_k)$ be the observed sample data.

$$
\ell(\boldsymbol{\gamma}) = \ln \ell(\boldsymbol{\gamma})
$$

$$
\begin{aligned}
&= \ln\left(\prod_{\mathbf{v}\in\mathbf{V}}\prod_{i=1}^{n}\frac{\gamma_i^{v_i}}{1+\gamma_i}\right)\\
&= \sum_{\mathbf{v}\in\mathbf{V}}\sum_{i=1}^{n}\ln\frac{\gamma_i^{v_i}}{1+\gamma_i}\\
&= \sum_{\mathbf{v}\in\mathbf{V}}\sum_{i=1}^{n}\left[v_i\ln\left(\gamma_i\right)-\ln\left(1+\gamma_i\right)\right].
\end{aligned}
$$

Now we calculate the partial derivative with respect to one coordinate of the parameters, $\gamma_i$, and set it to 0.

$$
\begin{aligned}
& & 0 &= \frac{\partial}{\partial\gamma_i}\ell(\boldsymbol{\gamma})\\
&\Leftrightarrow & 0 &= \sum_{\mathbf{v}\in\mathbf{V}}\left[\frac{v_i}{\gamma_i}-\rho_i\right]\\
&\Leftrightarrow & 0 &= \frac{1}{\gamma_i}\sum_{\mathbf{v}\in\mathbf{V}}v_i-k\rho_i\\
&\Leftrightarrow & \gamma_i\rho_i &= \frac{1}{k}\sum_{\mathbf{v}\in\mathbf{V}}v_i\\
&\Leftrightarrow & \rho_i &= 1-\frac{1}{k}\sum_{\mathbf{v}\in\mathbf{V}}v_i,
\end{aligned}
$$

and it can be expanded to all values of the vectors, returning

$$\widehat{\boldsymbol{\rho}} = \mathbb{1}_n - \frac{1}{k}\cdot\mathbf{V}\cdot\mathbb{1}_k.$$

So from the sample data we can estimate a $\boldsymbol{\rho}$ probabilities vector and by replacement we can also estimate the weights vector, and

$$\widehat{\gamma}_i = \frac{\sum_{\mathbf{v}\in\mathbf{V}}v_i}{k-\sum_{\mathbf{v}\in\mathbf{V}}v_i}.$$

Although, it's easy to notice that if $\sum_{\mathbf{v}}v_i = k$ or $\sum_{\mathbf{v}}v_i = 0$, then $\gamma_i$ is not well defined. Moreover, only if using those restrictions it's possible to show that the Hessian matrix $\mathcal{H}(\widehat{\boldsymbol{\gamma}})$ is negative semi-definite, proving that the root of the derivative is a local maximum. To handle this exception, we can use additive smoothing to the $\gamma$ values, and

$$\widehat{\boldsymbol{\gamma}}_{\alpha\text{-smoothed}} = \left(\mathbf{V}\cdot\mathbb{1}_k+\alpha\cdot\mathbb{1}_n\right)\circ\left((k+\alpha n)\cdot\mathbb{1}_n-\mathbf{V}\cdot\mathbb{1}_k\right)^{\circ-1}.$$

Having both the $\widehat{\boldsymbol{\rho}}$ and $\widehat{\boldsymbol{\gamma}}_{\alpha\text{-smoothed}}$ vectors is useful, because the first can be easly used to update the parameter values, using Bayesian inference via Beta distributions as priors, and the second handles exceptions and provides a more readable measurement of importance of each task for the overall routine schedule.

### 4.2 Model Predictions

## 5 Intelligent Tutoring System

But how will the mixture model and the tendency and routine distributions schedule practice routines for a student? The current method of SIACUA already uses bayesian networks to create a solid way to estimate a student's progress, but there isn't any approach to formulating schedules. Furthermore, the beliefs are just used as a prior for the values of the learning curves and are not sufficient to predict whether or not the student has mastered a certain topic.

So, the goal of the ITS it to gather information about the beliefs and the parameters of the regressions in (1) and schedule practice routines for the students, based on suggestions and spontaneous feedback. Moreover, since various topics are taken into account, the algorithm (5) has to be used in another context, to provide more individual results about each student.

## 5.1 Variables

Every time a student answers a question, the ITS should calculate and store a series of variables that will help it deciding what to recommend to the student. There are two types of variables: *static variables* and *topic related variables*. The first ones are:

- $T_{tutor}$ — tendency to follow the ITS' recommendations. Let $t \in \{0, 1\}$ be the outcome for a new recommendation that pops up. Considering the vector $\mathbf{t}$ of the outcomes, each $t_i$ is a realization of a random variate $T_i \sim \mathcal{B}(p)$, with $0 < p < 1$, and because the newest values are more significant ($t_1$ is newer than $t_2$), if $X_p \sim \text{Geo}(p)$, then

$$T_{tutor} = T(p) = \sum_{i=1}^{n} t_i \cdot \mathbb{P}(X_p = i).$$

  The value $p$ can initially be 0.5 (because at first the student either follows the ITS or not) and the value could be kept constant or be updated with bayesian inference, based on the number of times the student has followed the ITS's recommendations.

- $n_s$ — number of questions answered during the current session. Since this value ranges from 0 to $+\infty$, it's normally interpreted as a Poisson distribution $N \sim \text{Pois}(\lambda_s)$, where $\lambda_s$ is the average of the number of questions answered each session, that is stored and updated everytime the session ends. It is used its cumulative distribution function to calculate the critical value for which the result is greater than $c$:

$$\mathbb{P}(N \leq n_c) = \sum_{i=0}^{n_c} \mathbb{P}(N = i),$$
$$\mathbb{P}(N \leq i - 1) < c \leq \mathbb{P}(N \leq i) \Rightarrow n_c = i.$$

- $t$ — current time spent on a single question, measured in seconds. The ITS will also keep record of the mean value $\mu_t$, the mean of the squares $\mu_{t^2}$ and the sample standard deviation $\sigma_t$, updating it everytime a new value of $t$ is stored:

$$\begin{aligned} \mu_t' &= \frac{n_s \mu_t + t}{n_s + 1} \\ \mu_{t^2}' &= \frac{n_s \mu_{t^2} + t^2}{n_s + 1} \\ \sigma_t &= \sqrt{\mu_{t^2} - \mu_t^2}. \end{aligned}$$

  This way, since the variate that measures time is $T \sim \mathcal{N}(\mu_t, \sigma_t^2)$,

$$Z_t = \frac{T - \mu_t}{\sigma_t} \sim \mathcal{N}(0, 1).$$

  To find the critical value from which the cumulative distribution function has values above $c$, we calculate

$$\begin{aligned} & \mathbb{P}(Z_t \leq z_c) \geq c \\ \Leftrightarrow \quad & \mathbb{P}\left(\frac{T - \mu_t}{\sigma_t} \leq z_c\right) \geq c \\ \Leftrightarrow \quad & \mathbb{P}\left(T \leq z_c \sigma_t + \mu_t\right) \geq c \\ \Leftrightarrow \quad & z_c \sigma_t + \mu_t = t_c. \end{aligned}$$

Apart from the first variables, the other *static variables* are reseted once the current session ends. The *topic related variables* are:

- $T_{miss}$ — tendency to make a mistake on a question. The calculations made are the same as the ones for $T_{tutor}$. Considering the error vector $\mathbf{e}$ and $X \sim \text{Geo}(p)$, for $0 < p < 1$,

$$T_{miss} = T(p) = \sum_{i=1}^{n} e_i \cdot \mathbb{P}(X = i).$$

  The value of $p$ could be kept constant with 0.75 (because it's the default probability of answering incorrectly to a question) or it can be derived from the values of the student's learning curve.

- $n$ — number of questions answered. Because this variable is relative to each topic, we don't need to assign a certain distribution because it will just define the shape of our later distributions.
- $t_q$ — current time spent on a single question, measured in seconds. The same variables as in $t$ are stored and we're left with the distribution

  $$Z_{t_q} = \frac{T_q - \mu_{t_q}}{\sigma_{t_q}} \sim \mathcal{N}(0,1),$$

  and the critical value is also retrieved with

  $$\begin{aligned} & \mathbb{P}(Z_{t_q} \leq z_c) \geq c \\ \Leftrightarrow \qquad & z_c \sigma_{t_q} + \mu_{t_q} = (t_q)_c. \end{aligned}$$

- $S$ — success rate. It's a ratio between the number of correct answers by the number of questions answered. The *lazy* value for the variable would be plain division between the number of right answers and $n$, but since time has also an important role on the success rate, a more accurate value would be $(1-p)$, where $p$ is the value assigned when calculating $T_{miss}$. The critical value for the success $p_c$ is the value that represents stagnation in progress.
- $b$ — current belief, retrieved from the bayesian network. This values are already computed by the web application but the ITS stores them for convenience. If we regard $B \sim \text{Beta}(\alpha, \beta)$ as a *clumsy* variate that models beliefs, where $\alpha$ and $\beta$ are the number of right and wrong answers, respectively, then as $\alpha$ and $\beta$ grow large,

  $$\begin{aligned} B &\overset{a}{\sim} \mathcal{N}\left(\frac{\alpha}{\alpha+\beta}, \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}\right) \\ Z_b &= \frac{B - \frac{\alpha}{\alpha+\beta}}{\sqrt{\frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)}}} \sim \mathcal{N}(0,1). \end{aligned}$$

  If we assign $\alpha + \beta = n$, then $\alpha \approx nb$ and

  $$\begin{aligned} \frac{\alpha}{\alpha+\beta} &\approx \frac{nb}{n} = b, \\ \frac{\alpha\beta}{(\alpha+\beta)^2(\alpha+\beta+1)} &\approx \frac{nb(n-nb)}{n^2(n+1)} = \frac{b(1-b)}{n+1}. \end{aligned}$$

  and with the same calculations with the time variables, the critical value would be

  $$\begin{aligned} & \mathbb{P}(Z_b \leq z_c) \geq c \\ \Leftrightarrow \qquad & b + z_c \sqrt{\frac{b(1-b)}{n+1}} = b_c. \end{aligned}$$

- $b_n$ — average value of the beliefs from a topic and its neighbours. This variable follows the same logic as $b$ for the normal distribution.

## 5.2 Decisions

After computing and storing all the variables, the ITS performs additional calculations to find if it needs to recommend something to the student. To accomplish this, it corresponds each study suggestion with a variate such that all variates follow the same kind of distribution. Since we are dealing with multiple parameters and the ITS should only recommend something after crossing some threshold value, it is used the Generalized Extreme Value distribution for every tip.

In a distribution of the type $A \sim \text{GEV}(\mu, \sigma, \xi)$, the first parameter $\mu$ represents the location, the value from which the model starts recognizing that some action shoud be taken. The second, $\sigma$, it's the scale parameter, turning the curve wider as it gets bigger. The last one, $\xi$, it's the shape parameter, that defines the structure of the tail. On this context, we want only positive values (exceptionally 0, also), noticing that the curve gets more steep near $\mu$ as $\xi$ increases.

The 6 suggestions are:

- *Learn this topic.* You have little knowledge about it and have answered almost no questions.

$$\begin{aligned} A_{learn} &\sim \text{GEV}\left(b_c, \sqrt{\frac{b(1-b)}{n+1}}, n\right) \\ \mathbb{P}_{learn} &= \mathbb{P}\left(A_{learn} \leq 1-b\right) \end{aligned}$$

- *Do a quick scan.* You are spending too much time answering these questions and you are answering incorrectly most of the questions.

$$\begin{aligned} A_{scan} &\sim \text{GEV}\big((t_q)_c, \sigma_{t_q}, S\big) \\ \mathbb{P}_{scan} &= \mathbb{P}\left(A_{scan} \leq t_q\right) \end{aligned}$$

- *Revise these concepts.* You are making too many mistakes and you have little knowledge about that topic and neighbouring topics.

$$\begin{aligned} A_{revise} &\sim \text{GEV}\left((b_n)_c, \sqrt{\frac{b_n(1-b_n)}{n+1}}, T_{miss}\right) \\ \mathbb{P}_{revise} &= \mathbb{P}\left(A_{revise} \leq 1-b_n\right) \end{aligned}$$

- *Deepen this topic.* I see you listen to my recommendations but you still aren't making any progress whatsoever.

$$\begin{aligned} A_{deepen} &\sim \text{GEV}\Big(p_c, \sqrt{S(1-S)}, 1-T_{tutor}\Big) \\ \mathbb{P}_{deepen} &= \mathbb{P}\left(A_{deepen} \leq S\right) \end{aligned}$$

- *Focus on this topic.* Inspite of answering to a lot of questions about this, you seem to have low knowledge about it.

$$\begin{aligned} A_{focus} &\sim \text{GEV}\left(b_c, \sqrt{S(1-S)}, \frac{1}{n}\right) \\ \mathbb{P}_{focus} &= \mathbb{P}\left(A_{focus} \leq 1-b\right) \end{aligned}$$

- *Time to take a break.* I see you are spending too much time in every question and you have already answered a lot of questions in this session.

$$\begin{aligned} A_{break} &\sim \text{GEV}\left(t_c, \sigma_t, \frac{1}{n_s}\right) \\ \mathbb{P}_{break} &= \mathbb{P}\left(A_{break} \leq t\right) \end{aligned}$$

After calculating all the probabilities, we perform a $t$-test on the maximum probability, where the significance level is determined by the beliefs, because the higher the knowledge level in some topic, the less the need of practice scheduling.

$$\begin{aligned} \mathbb{P} &= \left\{\mathbb{P}_{learn}, \mathbb{P}_{scan}, \mathbb{P}_{revise}, \mathbb{P}_{deepen}, \mathbb{P}_{focus}, \mathbb{P}_{break}\right\}, \\ M_{\mathbb{P}} &= \max\{\mathbb{P}\}, \\ \mathbb{P}' &= \mathbb{P}\backslash M_{\mathbb{P}}, \\ U &= \overline{\mathbb{P}'} + t_{4,b}\frac{s_{\mathbb{P}'}}{\sqrt{5}}. \end{aligned}$$

If $M_{\mathbb{P}} > U$, then we must reject the null hypothesis and conclude that the action associated with $M_{\mathbb{P}}$ should be recommended, with a significance level of $b$.

### 5.3 Tutoring Curves

The only variable that's missing its calculation is the variable $c$ for the critical value. That's when the tutoring curves take place, using the same algorithm (5). However, instead of using data from all students, just the last 10 tests from the student that will perform the next test are taken into account. We initiate the algorithm with three components:

1. a primary component, with random values for $\vartheta_k$, but where the prior for $\alpha_k$ and $\beta_k$ are the ones calculated by the *clumsy* variate that models the beliefs, using the current belief.
2. a *lucky* component, with random values for $\vartheta_k$, $\alpha_k = 0$ and $\beta_k$ is equal to the number of questions answered per test.
3. an *unlucky* component, with random values for $\vartheta_k$, $\beta_k = 0$ and $\alpha_k$ is equal to the number of questions answered per test.

By the properties of the algorithm (5), the *lucky* and *unlucky* components model the extreme case where the student could answer all questions right and wrong, respectively, providing curves with very low and very high probabilities. After answering the questions of the new test, the ITS would then apply the results onto the mixture model to find the curve where the student fits the most, recording the percentage of the likelihood of the primary component (represented by $\mathbb{L}$), relative to the others.

$$\mathbb{P}(\mathbf{e}) = \sum_{k \in \{primary, lucky, unlucky\}} \rho_k \cdot \varpi_r \left( \boldsymbol{\mathcal{B}} \left( \mathbf{Q}^{\mathrm{T}} \cdot \boldsymbol{\vartheta}_k, \mathbf{e} \right) \right)$$

$$\mathbb{L} = \frac{\rho_{primary} \cdot \varpi_r \left( \boldsymbol{\mathcal{B}} \left( \mathbf{Q}^{\mathrm{T}} \cdot \boldsymbol{\vartheta}_{primary}, \mathbf{e} \right) \right)}{\mathbb{P}(\mathbf{e})}.$$

The variable $\mathbb{L}$ measures how consistent the student's practice routines are, so the lower the value, the lower is the confidence that the test is truthful. If the primary tutoring curve is too close to the lucky or unlucky tutoring curves, then we discard the one closest to the primary one.

Then, using exponential or logarithmic regressions, the values for $0 < a < 1$ and $0 < b' < 1$ are calculated to get the critical value $c$. We place the point $(a, b')$ in the rectangle $R = ]0,1[ \times ]0,1[$ and compute the distance from the point to the corner $(1,1)$. Although, we need to account for $\mathbb{L}$, as the lower it is, the close to the center of the rectangle the points goes. With a bit of vector algebra,

$$\begin{aligned}
& & (x, y) &= \left( a + (1 - \mathbb{L}) \left( \frac{1}{2} - a \right), b' + (1 - \mathbb{L}) \left( \frac{1}{2} - b' \right) \right) \\
&\Leftrightarrow & (x, y) &= \left( \frac{1}{2} + \mathbb{L} \left( a - \frac{1}{2} \right), \frac{1}{2} + \mathbb{L} \left( b' - \frac{1}{2} \right) \right) \\
&\Leftrightarrow & (x, y) &= \frac{1}{2} \left( 1 + \mathbb{L}(2a - 1), 1 + \mathbb{L}(2b' - 1) \right)
\end{aligned}$$

From that, we normalize the distance and

$$c = \frac{\|(x, y) - (1, 1)\|}{\sqrt{2}}.$$

This means that the lower the distance, the lower the critical value for the threshold for the ITS' recommendations. This can produce very high results to all decision thresholds but since we are performing $t$-tests to ponder which suggestion to make, it just prompts the ITS to schedule practice routines more regularly.

## 6 Notifications and Constructing Feedback

Apart from the 6 main suggestions, the ITS should also be able to provide notifications and constructive (either positive and negative) feedback in real-time. While the notifications are shown before the test, feedback is shown during each test and not at the end, and uses different criteria for possible suggestions.

Evey time a student clicks on a notification or answers a question, the ITS learns something about that student and, based on that information, provides other notifications and feedbacks that enhances his/her study performance. We call this the *reward*, and everytime the ITS is prompted to give notifications and feedback, it calculates the one with the highest reward.

## 6.1 Notifications

Notifications will just pop up on the main page of the web application. They are clickable links that redirect the user to some page or exercise. Every time the student logs on, the notification with highest reward at the moment will appear. Subsequent visits to the main page will not always produce notifications.

### 6.1.1 Reward

Every notification can be clicked or ignored, and after being clicked it can either result in a success or in a failure. So, for every notification $t$ we have the set of outcomes $k_t = \{0, 1\}$ for the click and $r_t = \{0, 1\}$ for the success. We record the collection of all sets as $H_t^s$, or *history* of $t$, and since $r_t$ can only be measured if $k_t = 1$, every time that $k_t = 0$ then $r_t = 0$ (you could also think like $r_t \Rightarrow k_t$). Then, the list $H_t^s$ contains only elements of the set $\mathcal{P} = \{(0, 0), (1, 0), (1, 1)\}$.

Based on this sets of outcomes and on sufficent data, it is possible to calculate $T_k^s$ and $T_r^s$, as it is with $T_{tutor}$, having then both the tendency values and the probabilities, $p_k^s$ and $p_r^s$, of either clicking or succeeding, that can be updated with bayesian inference. This sets a *policy* (represented as $\xi_t^s$) for the reward of each student individually, such that $\xi_t^s\,(t \mid H_t^s)$ is the probability of choosing the notification $t$, given the history of $t$, under the behaviour policy $\xi_t^s$.

We say a notification is elegible when it meets certain criteria, like test-streaks and belief values.

### 6.1.2 List of Notifications

- *You are missing out on <topic>.*
- *Continue your streak on <topic>.*
- *Have you studied <topic> today?*
- *<Random question about one topic>.*
- *Haven't seen you in a long time, <name>!*
- *Hello again, <name>! Let's take on where we left off last time?*

## 6.2 Feedback

### 6.2.1 Reward

### 6.2.2 List of Feedbacks

- *Level Up!* (positive).
- *Level Down...* (negative).
- *Nice streak! Keep it up!* (positive).
- *You missed a few in a row, take your time to answer...* (negative).
- *You're nailing it! Just a few more to go!* (positive)
- *Don't give up, just a few more to go!* (negative)

## 6.3 Test Layout

Each test consists on a number $N$ of questions of progressing difficulties. Consecutive right and wrong answers lead to going up or down levels of difficulty, respectively. After each question, the ITS computes the feedback with highest reward and provides it. The

# 7 Programming

Since SIACUA is programmed in C$\sharp$, the ITS will also be constructed in that language, as a Class Library (.dll file), to be then used as a reference for this and other web applications.

# References

[1] Benjamin Clement, Didier Roy, Pierre-Yves Oudeyer, and Manuel Lopes. Online optimization of teaching sequences with multi-armed bandits. In *Proceedings of the 7th International Conference on Educational Data Mining*, 2014.

[2] Luís Descalço and Paula Carvalho. Siacua. https://siacua.web.ua.pt. Accessed: 2023-11-22.

[3] Meltem Eryılmaz and Afaf Adabashi. Development of an intelligent tutoring system using bayesian networks and fuzzy logic for a higher student academic performance. *Applied Sciences*, 10(19), 2020.

[4] Hugo Gamboa and Ana L. N. Fred. Designing intelligent tutoring systems: A bayesian approach. In *International Conference on Enterprise Information Systems*, 2001.

[5] Alan Ramírez-Noriega, Reyes Juárez-Ramírez, and Yobani Martínez-Ramírez. Evaluation module based on bayesian networks to intelligent tutoring systems. *International Journal of Information Management*, 37(1, Part A):1488–1498, 2017.

[6] Matthew Streeter. Mixture modeling of individual learning curves. In *Proceedings of the 8th International Conference on Educational Data Mining*, 2015.

[7] Kevin P. Yancey and Burr Settles. A sleeping, recovering bandit algorithm for optimizing recurring notifications. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, KDD '20, pages 3008–3016, New York, NY, USA, 2020. Association for Computing Machinery.