Learning Attacker Utilities in Nested-Logit Security Games

Chun Kai Ling[*]
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
chunkail@andrew.cmu.edu

Hoon Oh[†]
Department of Computer Science
Carnegie Mellon University
Pittsburgh, PA 15213
oh94kr@ymail.com

Abstract

Utility functions of attackers in Stackelberg security games may be learned by repeated observation of their response to 3 different defender strategies. However, existing work makes strong assumption on their behavioral models. We develop a new approach based on the nested-logit choice models, a generalization of the logit quantal response. We provide a scheme for learning attacker utilities under these models and provide analysis showing a similar convergence rate to existing work. Furthermore, an approach for learning attacker rationality parameters is proposed and tested experimentally with synthetic data.

1 Learning Attacker Payoffs

Stackelberg security games as a modelling tool has gained much popularity in recent years, with many practical successes in wildlife conservation and airport security. Until recently, the bulk efforts have been spent in modelling accurate behavioral models for agents *assuming their payoffs are known*. The assumption that payoffs are known is often made on the basis that the researcher has access experts or oracles with domain knowledge, or when specific game structures are assumed (e.g. zero-sum games). None of these premises are satisfied in real-world applications, particularly in security domains.

As such, recent years have seen a renewed focus on *learning* games. Earlier work do not attempt to learn player payoffs and instead seek to learn strategies which perform well [4, 8, 2]. This is a (somewhat) easier task, since knowing the payoffs of other players implies a better response. However, in many cases, we do actually want to learn the player's payoffs. For example, in security games, we may want to understand the features motivating or attracting an adversary, rather than just their observed strategy. This helps the defender better understand how aspects of the game can be manipulated or changed to get a desirable outcome.

In this project, we adopt a framework similar to Haghtalab et. al. [3]. In this framework, it is assumed that the attacker payoff is some linear function of the defender's coverage probability. The defender has the advantage of commitment, and the attacker responds according to the logit quantal response [6, 5]. The goal is to learn the *attacker's utility as a a function* of the defender's coverage, by observing samples from the attackers response. In a typical security game, the attacker obtains the defender's mixed strategy by repeatedly observing the defender's actions. This process is fairly time consuming, hence it is crucial that the number of 'tests' be kept as few as possible. It was shown in [3] that the defender has to obtain sufficient samples from just 3 different coverage probabilities before learning attacker utility functions with high probability.

It is well known that the assumption of logit quantal response is equivalent to a random utility model (RUM) which adds i.i.d. gumbel distributed noise to each choice (also known as alternative). The

[*] Course project for CMU Graduate AI, 15-780, Spring 2018

[†] Course project for CMU Graduate AI, 15-780, Spring 2018

independence assumption is known to be a source of pathological cases, especially when there are alternatives with similar utilities, but are ‘essentially similar’ apart from their labels [1, 6]. We point the reader to [7] for an easily accessible overview of choice models.

The *nested-logit choice model* (NL) is a generalization of the standard logit and avoids some of these pathological cases by assuming a hierarchical structure (or clustering) of the alternatives, where *within* clusters at the same level (sharing the same parent), a separate logit quantal choice is made based on payoffs of their children. In the context of security games, we may group targets of a similar nature (e.g. airports versus metro) together. Unlike the logit quantal model, noise across groups *are* correlated (but are i.i.d within each group). Mathematically (to be explained later), this corresponds to *rationality parameters* varying between each group.

In our work, we extend the work in [3] to the NL model. First, we derive closed form estimators for attacker utilities and show that given knowledge of rationality parameters, the number of samples (or convergence rate) required to learn attacker utilities with high probability is identical to [3] asymptotically. Second, we derive closed form expressions to learn the rationality parameters. We also provide some preliminary analysis on the convergence rates in this case. The second contribution is somewhat novel and interesting – in fact, it is known that for nested logits, learning the rationality parameters *alone* (assuming knowledge of true utilities) via maximum-likelihood-estimation yields a non-convex problem. This is a consequence of the flexibility afforded by adopting the framework of [3].

2 Nested logit models

This section briefly introduces NL models, for a more in-depth discussion, refer to [7]. For simplicity, assume that there are η nests (groups) and n targets in within each group, making for $\eta \times n$ targets [3]. Each target yields an expected utility of $u_{\tau,t}(x_{\tau,t}) = w_{\tau,t} x_{\tau,t} + c_{\tau,t}$, where $x_{\tau,t}$ where τ and t are the target's nest and target respectively, $\mathbf{x}$ is a vector containing defender coverage probabilities, $\mathbf{w}, \mathbf{c}$ are vectors containing the coefficients for the linear utility model.

For this paragraph, assume that coverage probabilities $\mathbf{x}$ is fixed. The probability that the attacker chooses target (τ, t) given that *some* target in (τ) is selected follows a standard softmax.

$$D^{\mathbf{x}}(\tau, t)/D^{\mathbf{x}}(\tau) = \frac{\exp(\gamma_\tau u_{\tau,t}(x_{\tau,t})}{\sum_{j\in\tau} \exp\left(\gamma_\tau u_{\tau,j}(x_{\tau,j})\right)}, \tag{1}$$

where $D^{\mathbf{x}}$ is the attackers response given defender coverage $\mathbf{x}$, $D^{\mathbf{x}}(\tau, t)$ is the probability that target t in τ is attacker, $D^{\mathbf{x}}(\tau)$ is the probability that some target in nest τ is attacker (i.e. $D^{\mathbf{x}}(\tau) = \sum_{j\in\tau} D^{\mathbf{x}}(\tau, j)$, where we abuse the notation $j \in \tau$ to mean all targets t within nest τ. The parameter γ_τ only depends on the nest τ and is known as the rationality parameter. Observe that when $\gamma_\tau \to \infty$, the softmax operation tends to the standard argmax operation (full rationality). Conversely, if γ_τ tends to 0, then objects within this nest are chosen with near equal probability (complete irrationality). It is well known that to be RUM, $\gamma \in [1, \infty]$. Further, in the NL model, we have

$$D^{\mathbf{x}}(\tau) = \frac{\exp\left(\frac{1}{\gamma_\tau} \log \sum_{j\in\tau} \exp\left(\gamma_\tau u_{\tau,j}(x_{\tau,j})\right)\right)}{\sum_{k\in\text{nests}} \exp\left(\frac{1}{\gamma_k} \log \sum_{j\in k} \exp\left(\gamma_k u_{k,j}(x_{k,j})\right)\right)} \tag{2}$$

This may be interpreted as applying a standard logit choice model with rationality 1 to each of the nests, where the value of each nest is the log-sum-exp of that nest's children's utilities (i.e. approximating the maximum in the nest, which is what log-sum-exp does). The probabilities $D^{\mathbf{x}}(\tau, t)$ may be obtained by multiplying the above 2 expressions together. As a sanity check, observe that if all γ's were 1, we would recover the standard logit model with $\eta \times n$ targets.

To align ourselves closer to the work of [3], we will ‘normalize’ the utilities in each nest by subtracting the constant term in the *last* (n-th) target to all targets in the nest ($c_{\tau,n}$), and instead re-add this value to the nest (outside of the log-sum-exp and factor of $\frac{1}{\gamma_\tau}$. Intuitively, this does nothing to change the outcome of the decision making process, as we are merely rewarding the attacker for

[3] The assumption of similarly sized groups is for simplicity in exposition,our analysis easily extends to groups of uneven sizes

simply ‘choosing’ a certain nest (and deducting that from the future payoffs). We denote this value in shorthand as c_τ. This gives us the following expression in normalized form:

$$D^{\mathbf{x}}(\tau) = \frac{\exp\left(c_\tau + \frac{1}{\gamma_\tau}\log\sum_{j\in\tau}\exp\left(\gamma_\tau u_{\tau,j}(x_{\tau,j})\right)\right)}{\sum_{k\in\text{nests}}\exp\left(c_k + \frac{1}{\gamma_k}\log\sum_{j\in k}\exp\left(\gamma_k u_{k,j}(x_{k,j})\right)\right)} \tag{3}$$

3 Learning Attacker Utilities in NL Models

Our main result is almost identical to Theorem 3.1 of [3], which we reproduce partially here for completeness.

Theorem 1. *Suppose the utility functions $u(\cdot)$ are linear. Suppose the defender choose 3 coverage proabilites $\mathbf{p}, \mathbf{q}, \mathbf{r}$ such that for any $t < n$, and for every nest τ, $|(p_{\tau,t} - q_{\tau,t})(p_{\tau,n} - r_{\tau,n})| - (p_{\tau,n} - q_{\tau,n})(p_{\tau,t} - r_{\tau,t})| \geq \lambda$, and for any 2 strategies $\mathbf{x}, \mathbf{y} \in \{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$, $|x_{\tau,t} - y_{\tau,t}| \geq \nu$. Further, assume that for all of the 3 strategies, all attacker targets are played with probability greater than ρ. Then, having access to $m = \Omega\left(\frac{1}{\rho}\left(\frac{1}{\epsilon\nu\lambda}\right)^2\log(n/\delta)\right)$ samples of attacker responses to each of the 3 strategies implies that with probability $1-\delta$, one may, for all τ, t, learn $u_{\tau,t}(\cdot)$ up to error ϵ, for all possible $\mathbf{x}$ (including those not in $\{\mathbf{p}, \mathbf{q}, \mathbf{r}\}$, also known as uniformly learning u).*

The rest of the report is dedicated to the method of estimating the parameters w, c. The proofs of convergence are in the Appendix.

3.1 Within-nest relative utilities

This key to this result in both our work [3] is the property of *independence from irrelevant alternatives* (IIA). This means that for some fixed nest (in the case of the logit, all targets belong to one big nest), the *ratio* attacker probabilities for two targets t_1, t_2 does *not* change if the utility of some other target $t3$ in the same nest is changed (i.e. $t3$ is a irrelevant alternative). Note that IIA does *not* hold across nests. This property may be easily verified by the expressions in the previous section.

For any defender’s coverage $\mathbf{p}$, applying the IIA property to targets t, n within nest τ gives us the following relation

$$\gamma_\tau u_{\tau,t}(p_{\tau,t}) = \log\left(\frac{D^{\mathbf{p}}(\tau,t)}{D^{\mathbf{p}}(\tau,n)}\right) + \gamma_\tau u_{\tau,n}(p_{\tau,n}) \tag{4}$$

Combining the assumption of linear models, $u_{\tau,t}(p_{\tau,t}) = w_{\tau,t}p_{\tau,t} + c_{\tau,t}$ with relation (4) for the second defender strategy $\mathbf{q}$ allows for the elimination of the term $c_{\tau,t}$, giving

$$\gamma_\tau w_{\tau,t}\left(p_{\tau,t} - q_{\tau,t}\right) = \log\left(\frac{D^{\mathbf{p}}(\tau,t)}{D^{\mathbf{p}}(\tau,n)}\right) - \log\left(\frac{D^{\mathbf{q}}(\tau,t)}{D^{\mathbf{q}}(\tau,n)}\right) + \gamma_\tau w_{\tau,n}\left(p_{\tau,n} - q_{\tau,n}\right). \tag{5}$$

Replacing $\mathbf{q}$ with the third converage vector $\mathbf{r}$ allows us to represent $w_{\tau,t}$ purely in terms of the D’s, p’s, q’s and r’s and γ’s

$$\gamma_\tau w_{\tau,n} = \frac{(p_{\tau,t} - r_{\tau,t})\log\left(\frac{D^{\mathbf{p}}(\tau,t)D^{\mathbf{q}}(\tau,n)}{D^{\mathbf{q}}(\tau,t)D^{\mathbf{p}}(\tau,n)}\right) - (p_{\tau,t} - q_{\tau,t})\log\left(\frac{D^{\mathbf{p}}(\tau,t)D^{\mathbf{r}}(\tau,n)}{D^{\mathbf{r}}(\tau,t)D^{\mathbf{p}}(\tau,n)}\right)}{(p_{\tau,t} - q_{\tau,t})(p_{\tau,n} - r_{\tau,n}) - (p_{\tau,n} - q_{\tau,n})(p_{\tau,t} - r_{\tau,t})} \tag{6}$$

$$\gamma_\tau w_{\tau,t} = \frac{\log\left(\frac{D^{\mathbf{p}}(\tau,t)}{D^{\mathbf{p}}(\tau,n)}\right) - \log\left(\frac{D^{\mathbf{q}}(\tau,t)}{D^{\mathbf{q}}(\tau,n)}\right) + \gamma_\tau w_{\tau,n}\left(p_{\tau,n} - q_{\tau,n}\right)}{p_{\tau,t} - q_{\tau,t}} \tag{7}$$

$$\gamma_\tau c_{\tau,t} = \log\left(\frac{D^{\mathbf{p}}(\tau,t)}{D^{\mathbf{p}}(\tau,n)}\right) + \gamma_\tau w_{\tau,n}p_{\tau,n} - \gamma_\tau w_{\tau,t}p_{\tau,t} \tag{8}$$

3.2 Across-nest relative utilities

Denote $D^{\mathbf{p}}(\tau)$ and $D^{\mathbf{p}}(t|\tau)$ to be the probabilities and conditional probabilities of selecting A) a target in nest τ and B) target t given some target in nest τ was chosen. For some fixed $D^{\mathbf{p}}\left(t|\tau\right)$, we may write the ‘pseudo’ utility of nest τ under coverage $\mathbf{p}$ by

$$u_\tau | D^{\mathbf{p}}\left(t|\tau\right)(\mathbf{p}) = c_\tau + \sum_t u_{\tau,t}(p_{\tau,t}) D^{\mathbf{p}}\left(t|\tau\right) + \frac{1}{\gamma_\tau} H\left(D^{\mathbf{p}}\left(t|\tau\right)\right) \tag{9}$$

$$= c_\tau + \sigma(\gamma_\tau, u_\tau) \tag{10}$$

where c_τ is the base payoff from this nest and $\sigma(\gamma_\tau, u_\tau(\mathbf{p})) = \frac{1}{\gamma_\tau}\log\left(\sum_t \exp\left(\gamma_\tau u_{\tau,t}(p_{\tau,t})\right)\right)$. Our objective is to obtain c_τ. For conciseness, we write the above expression as, $u_\tau(\mathbf{p})$. Applying IIA now to each of the nests,

$$u_\tau(\mathbf{p}) = \log\left(\frac{D^{\mathbf{p}}(\tau)}{D^{\mathbf{p}}(\eta)}\right) + u_\eta(\mathbf{p}) \tag{11}$$

If we knew all the γ’s a priori then we may fix c_η to be 0 (wlog), giving us $u_\eta(\mathbf{p}) = \sigma(\gamma_\eta, u_\eta(\mathbf{p})$. Rearranging c_τ for all $\tau < \eta$ gives

$$c_\tau = \log\left(\frac{D^{\mathbf{p}}(\tau)}{D^{\mathbf{p}}(\eta)}\right) - \sigma(\gamma_\tau, u_\tau(\mathbf{p})) + \sigma(\gamma_\eta, u_\eta(\mathbf{p})) \tag{12}$$

4 Learning rationality parameters γ

Being able to learning attacker payoff functions is unsurprising. Here, we present a method of learning the rationality parameters γ and analyze its rate of convergence (without knowledge of c_τ). With reference to (9) and (11).

$$\sigma(\gamma_\tau, u_\tau(\mathbf{p})) + c_\tau = \log\left(\frac{D^{\mathbf{p}}(\tau)}{D^{\mathbf{p}}(\eta)}\right) + \sigma(\gamma_\eta, u_\eta(\mathbf{p})) \tag{13}$$

Subtracting the expressions between $\mathbf{p}$ and $\mathbf{q}$ eliminates the unknown c_τ.

$$\sigma(\gamma_\tau, u_\tau(\mathbf{p})) - \sigma(\gamma_\tau, u_\tau(\mathbf{q})) = \log\left(\frac{D^{\mathbf{p}}(\tau) D^{\mathbf{q}}(\eta)}{D^{\mathbf{p}}(\eta) D^{\mathbf{q}}(\tau)}\right) + \sigma(\gamma_\eta, u_\eta(\mathbf{p})) - \sigma(\gamma_\eta, u_\eta(\mathbf{q})) \tag{14}$$

Rewriting,

$$\sigma(\gamma_\tau, u_\tau(\mathbf{p})) - \sigma(\gamma_\tau, u_\tau(\mathbf{q})) = \frac{1}{\gamma_\tau}\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{q})) \tag{15}$$

$$\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{q})) = \log\left(\frac{\sum_t \exp\left(\gamma_\tau u_{\tau,t}(p_{\tau,t})\right)}{\sum_t \exp\left(\gamma_\tau u_{\tau,t}(q_{\tau,t})\right)}\right) \tag{16}$$

Applying Equation (14) to $\mathbf{p}$ and $\mathbf{r}$ and rearranging gives

$$\frac{1}{\gamma_\eta} = \frac{\frac{1}{\gamma_\tau}\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{r})) - \log\left(\frac{D^{\mathbf{p}}(\tau) D^{\mathbf{r}}(\eta)}{D^{\mathbf{p}}(\eta) D^{\mathbf{r}}(\tau)}\right)}{\beta(\gamma_\eta, u_\tau(\mathbf{p}), u_\tau(\mathbf{r}))}. \tag{17}$$

Substituting $\frac{1}{\gamma_\eta}$ into (14) gives

$$\frac{1}{\gamma_\tau} = \frac{\log\left(\frac{D^{\mathbf{p}}(\tau) D^{\mathbf{q}}(\eta)}{D^{\mathbf{p}}(\eta) D^{\mathbf{q}}(\tau)}\right)\beta(\gamma_\eta, u_\eta(\mathbf{p}), u_\eta(\mathbf{r})) - \log\left(\frac{D^{\mathbf{p}}(\tau) D^{\mathbf{r}}(\eta)}{D^{\mathbf{p}}(\eta) D^{\mathbf{r}}(\tau)}\right)\beta(\gamma_\eta, u_\eta(\mathbf{p}), u_\eta(\mathbf{q}))}{\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{q}))\beta(\gamma_\eta, u_\eta(\mathbf{p}), u_\eta(\mathbf{r})) - \beta(\gamma_\eta, u_\eta(\mathbf{p}), u_\eta(\mathbf{q}))\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{r}))}. \tag{18}$$

Importantly, computing terms on the right-hand side do not require γ’s to be known (despite looking like it). This is because computing β only requires knowledge of the *product* γu, which may be computed using the equations in (6).

5 Experiments and discussion

Due to time constraints, our experimental results are preliminary and do *not* represent a fair evaluation of effectiveness; instead, they serve to verify our results. Our experimental setup is as follows. We

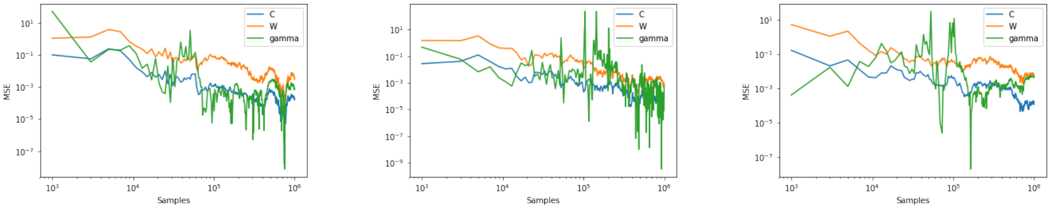

Figure 1: Experimental results depicting accuracy of parameters with number of samples. Note the log-log scale for 3 different runs (each run has the same parameters but with different sampled attacker strategies)

assuming 2 nests with 2 targets each (total of 4 targets). We may conveniently represent this in matrix form:

$$W = \begin{bmatrix} -2 & 3 \\ -1 & 1.5 \end{bmatrix} \qquad C = \begin{bmatrix} 0.3 & -0.2 \\ 0.6 & -0.5 \end{bmatrix}, \tag{19}$$

where targets rows represent targets in the same nest. Writing, P, Q, R as matrices containing defender coverage probabilities, we set

$$p = \begin{bmatrix} 0.25 & 0.25 \\ 0.25 & 0.25 \end{bmatrix} \qquad q = \begin{bmatrix} 0.4 & 0.2 \\ 0.2 & 0.2 \end{bmatrix} \qquad r = \begin{bmatrix} 0.15 & 0.2 \\ 0.3 & 0.35 \end{bmatrix}. \tag{20}$$

The payoff 'matrix' (not the same definition as a 2-player game!) may be written as $U^{\mathbf{p}} = P \circ W + C$, where $\circ$ is the Hadarmard product. We may do the same to obtain $U^{\mathbf{q}}$ and $U^{\mathbf{r}}$. We assume true $\gamma_1 = 1.5$, $\gamma_2 = 2$. Under the nested logit model, attacker strategies are computed and found to be:

$$D^{\mathbf{p}} = \begin{bmatrix} 0.1362 & 0.4195 \\ 0.3204 & 0.1239 \end{bmatrix} \qquad D^{\mathbf{q}} = \begin{bmatrix} 0.1041 & 0.4014 \\ 0.3800 & 0.1145 \end{bmatrix} \qquad D^{\mathbf{r}} = \begin{bmatrix} 0.1917 & 0.3492 \\ 0.2911 & 0.1680 \end{bmatrix} \tag{21}$$

We observed the MSE for the parameters W, C and γ. Note that W, C were computed assuming knowledge of γ. While MSE is not quite what a defender would be interested in (e.g. the worst case error in estimates of $u(x)$ over *all* x may be more appropriate), it still remains a reasonable proxy in that an MSE of 0 implies perfect learning of $u(x)$. The results for 3 different runs are presented in Figure 1.

In general, the results are somewhat acceptable. Observe that the estimates for γ are fairly poor as compared to the other C, W, suggesting that the convergence rates are much poorer. This stems from 2 reasons: first, the assuption that ϵ' is sufficiently low may not be satisfied even unless m is extremely large and if that is so, the denominator in (18) may go close to zero, causing a blowup in the estimation of γ.

We end off by commenting on the theoretical result when estimating w and c, assuming known γ's. It may seem odd that we can achieve the same convergence rate, 'independent' of γ, given that our model appears more complicated. For when the γ's are large, the attacker is more rational and there is more 'information' revealed in his actions, implying a faster convergence. Yet, we know that the whole reason why we may estimate utilities only for the QRE and *not* a perfectly rational attacker is *precisely* because of irrationality in the attacker (in the perfectly rational case, there will be some actions which may never be chosen). This may appear to be somewhat of a contractiction. In reality, there is a second counterbalancing effect in play – ρ, which is the lowest probability of playing some action. When $\gamma \to \infty$, $\rho \to 0$ (unless there are ties in utilities), which in turn pushes up the number of samples required. In other words, the bounds we derived are somewhat misleading, since the effect of γ is hidden within ρ.

References

[1] Gerard Debreu. *Individual choice behavior: A theoretical analysis*. 1960.

[2] John Fearnley et al. "Learning equilibria of games via payoff queries." In: *Journal of Machine Learning Research* 16 (2015), pp. 1305–1344.

[3] Nika Haghtalab et al. "Three Strategies to Success: Learning Adversary Models in Security Games". In: (2016).

[4] Joshua Letchford, Vincent Conitzer, and Kamesh Munagala. "Learning and approximating the optimal strategy to commit to". In: *International Symposium on Algorithmic Game Theory*. Springer. 2009, pp. 250–262.

[5] R Duncan Luce. *Individual choice behavior: A theoretical analysis*. Courier Corporation, 2005.

[6] Daniel L McFadden. "Quantal choice analaysis: A survey". In: *Annals of Economic and Social Measurement, Volume 5, number 4*. NBER, 1976, pp. 363–390.

[7] Kenneth E Train. *Discrete choice methods with simulation*. Cambridge university press, 2009.

[8] Yevgeniy Vorobeychik, Michael P Wellman, and Satinder Singh. "Learning payoff functions in infinite games". In: *Mach Learn* 67 (2007), pp. 145–168.

6 Appendix

6.1 Convergence rates for learning w, c, u

In reality, only random samples of attacker strategies are observed – these estimators are denoted by $\hat{(\cdot)}$; we thus need a bound on how much error is incurred. First, begin by giving a lower bound on 'good' events occurring. Similar to Naghtalab et. al (2015), for any fixed nest τ, we may apply Chernoff bounds to yield bounds on the error ratio $\frac{\hat{D}^{\mathbf{p}}(\tau,t)}{D^{\mathbf{p}}(\tau,t)}$. In the case of nested-logits, we also require the inequality to hold for ratios across each nest $\frac{\hat{D}^{\mathbf{p}}(\tau)}{D^{\mathbf{p}}(\tau)}$.

$$\mathbb{P}\left[\frac{1}{1+\epsilon} \le \frac{\hat{D}^{\mathbf{p}}(\tau,t)}{D^{\mathbf{p}}(\tau,t)} \le 1+\epsilon\right] \ge 1 - 2\exp\left(-mD^{\mathbf{p}}\left(\tau,t\right)\epsilon^2/4\right) \tag{22}$$

$$\mathbb{P}\left[\frac{1}{1+\epsilon} \le \frac{\hat{D}^{\mathbf{p}}(\tau)}{D^{\mathbf{p}}(\tau)} \le 1+\epsilon\right] \ge 1 - 2\exp\left(-mD^{\mathbf{p}}\left(\tau\right)\epsilon^2/4\right) \tag{23}$$

With at most n targets, there are at most n nests and at most $2n$ inequalities to be satisfied. Hence, setting $m = \Omega\left(\frac{1}{\rho\epsilon^2}\log(\frac{n}{\delta})\right)$ (as before) ensures that each of the inequalities is satisfied with probability $1 - \frac{\delta}{2n}$; taking the union bound gives a probability of $1 - \delta$.

Set $\epsilon' = \epsilon\lambda\nu/512$. The following bounds which hold with high probability follow (up to constant factors and that of γ_τ) directly from [3].

$$\gamma_\tau|w_{\tau,n} - \hat{w}_{\tau,n}| \le 8\epsilon'/\lambda \le \epsilon/64 \tag{24}$$

$$\gamma_\tau|w_{\tau,t} - \hat{w}_{\tau,t}| \le (4\epsilon' + 8\epsilon'/\lambda)/\nu \le 12\epsilon'/(\nu\lambda) \le \epsilon/32 \tag{25}$$

$$\gamma_\tau|c_{\tau,t} - \hat{c}_{\tau,t}| \le \epsilon'(2\nu\lambda + 8\nu + 12)/(\nu\lambda) \le 24\epsilon'/(\nu\lambda) \le \epsilon/8 \tag{26}$$

Lemma 2. *With high probability,* $|u_{\tau,t}(x_{\tau,t}) - \hat{u}_{\tau,t}(x_{\tau,t})| \le 36\epsilon'/(\nu\lambda\gamma_\tau) \le \frac{5}{32}\frac{\epsilon}{\gamma_\tau}$.

Proof. Using the linear definition of $u(\cdot)$ and $x_t \in [0, 1]$. □

Lemma 3. *Suppose $u_{\tau,t}$ may be uniformly learned within error ϵ/γ_τ, i.e. for any x, $|u_{\tau,t}(x) - \hat{u}_{\tau,t}(x)| \le \epsilon/\gamma_\tau$, then*

$$|\sigma(\gamma_\tau, u_\tau(\mathbf{p})) - \hat{\sigma}(\gamma_\tau, \hat{u}_\tau(\mathbf{p}))| \le \epsilon \tag{27}$$

Proof. By definition.

$$|\sigma(\gamma_\tau, u_\tau(\mathbf{p})) - \hat{\sigma}(\gamma_\tau, \hat{u}_\tau(\mathbf{p}))| = \frac{1}{\gamma_\tau}\left|\log\left(\frac{\sum_t \exp\left(\gamma_\tau \hat{u}_{\tau,t}(p_{\tau,t})\right)}{\sum_t \exp\left(\gamma_\tau u_{\tau,t}(p_{\tau,t})\right)}\right)\right| \tag{28}$$

$$\le \frac{1}{\gamma_\tau}\left|\log\left(\frac{\sum_t \exp\left(\gamma_\tau u_{\tau,t}(p_{\tau,t}) + \epsilon\right)}{\sum_t \exp\left(\gamma_\tau u_{\tau,t}(p_{\tau,t})\right)}\right)\right| \tag{29}$$

$$\le \epsilon/\gamma_\tau \le \epsilon \tag{30}$$

The first inequality holds since log-sum-exp is monotonic increasing. The lower bound proceeds in the exact same way with the negative ϵ term in the second line, the last line uses the assumption that $\gamma_\tau \ge 1$. □

We now seek to bound the error for c_τ.

$$|c_\tau - \hat{c}_\tau| \leq \left|\log\left(\frac{D^{\mathbf{p}}(\tau)\hat{D}^{\mathbf{p}}(\eta)}{D^{\mathbf{p}}(\eta)\hat{D}^{\mathbf{p}}(\tau)}\right)\right| + |\sigma(\gamma_\tau, u_\tau(\mathbf{p})) - \hat{\sigma}(\gamma_\tau, \hat{u}_\tau(\mathbf{p}))| + |\sigma(\gamma_\eta, u_\eta(\mathbf{p})) - \hat{\sigma}(\gamma_\eta, \hat{u}_\eta(\mathbf{p}))| \tag{31}$$

$$\leq 2\epsilon' + 5\epsilon/16 \leq \epsilon\lambda\nu/256 + 5\epsilon/16 \leq \epsilon(3/8) \tag{32}$$

The second line uses the definition of bounds on ratios and the fact that $\log(1+x) \leq x$, as well as Lemma 3, the third line follows from the definition of ϵ', the last line from the fact that $\lambda \leq 2, \nu \leq 1$. Our final bound shows uniform learning of utilities by combines the two bounds,

$$|(u_{\tau,t}(x_{\tau,t}) + c_\tau) - (\hat{u}_{\tau,t}(x_{\tau,t}) + \hat{c}_\tau)| \leq |u_{\tau,t}(x_{\tau,t}) - \hat{u}_{\tau,t}(x_{\tau,t})| + |c_\tau - \hat{c}_\tau| \tag{33}$$

$$\leq \frac{5}{32}\frac{\epsilon}{\gamma_\tau} + \frac{3\epsilon}{8} \tag{34}$$

$$\leq \frac{5\epsilon}{32} + \frac{3\epsilon}{8} = \frac{17\epsilon}{32} \leq \epsilon \tag{35}$$

6.2 Convergence rates for learning γ

We want to find an expression for $|\frac{1}{\gamma_\tau} - \frac{1}{\hat{\gamma}_\tau}|$, where $\hat{\gamma}_\tau$ is given by (18). We make an additional assumption that each of the (true, not sampled) β terms in (18) has absolute value no greater than θ, and that the absolute value of the denominator in (18) is greater than ϕ.

Lemma 4. $|ab - \hat{a}\hat{b}| \leq |a||\hat{b} - b| + |b||\hat{a} - a| + |\hat{a} - a||\hat{b} - b|$

Proof. Write $\hat{a} = a + (\hat{a} - a)$ and $\hat{b} = b + (\hat{b} - b)$ and expand. □

Lemma 5. *Suppose that all $u_{\tau,t}$ may be uniformly approximated to ϵ/γ_τ accuracy. Then*

$$\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{q})) - \hat{\beta}(\gamma_\tau, \hat{u}_\tau(\mathbf{p}), \hat{u}_\tau(\mathbf{q})) \leq 2\epsilon \tag{36}$$

Proof. Similar to that for Lemma 3.

$$\left|\log\left(\frac{\sum_t \exp(\gamma_\tau u_{\tau,t}(p_{\tau,t}))}{\sum_t \exp(\gamma_\tau u_{\tau,t}(q_{\tau,t}))}\right) - \log\left(\frac{\sum_t \exp(\gamma_\tau \hat{u}_{\tau,t}(p_{\tau,t}))}{\sum_t \exp(\gamma_\tau \hat{u}_{\tau,t}(q_{\tau,t}))}\right)\right| \tag{37}$$

$$\leq \left|\log\left(\frac{\sum_t \exp(\gamma_\tau \hat{u}_{\tau,t}(p_{\tau,t}))}{\sum_t \exp(\gamma_\tau u_{\tau,t}(p_{\tau,t}))}\right) - \log\left(\frac{\sum_t \exp(\gamma_\tau \hat{u}_{\tau,t}(q_{\tau,t}))}{\sum_t \exp(\gamma_\tau u_{\tau,t}(q_{\tau,t}))}\right)\right| \tag{38}$$

$$\leq \left|\log\left(\frac{\sum_t \exp(\gamma_\tau u_{\tau,t}(p_{\tau,t}) + \epsilon)}{\sum_t \exp(\gamma_\tau u_{\tau,t}(p_{\tau,t}))}\right)\right| + \left|\log\left(\frac{\sum_t \exp(\gamma_\tau u_{\tau,t}(q_{\tau,t}) + \epsilon)}{\sum_t \exp(\gamma_\tau u_{\tau,t}(q_{\tau,t}))}\right)\right| \leq 2\epsilon \tag{39}$$

The lower bound may be worked out the same way using $-\epsilon$ in the numerator. □

Combining this with Lemma 2, we know that with high probability, $|\beta(\gamma_\tau, u_\tau(\mathbf{p}), u_\tau(\mathbf{q})) - \hat{\beta}(\gamma_\tau, \hat{u}_\tau(\mathbf{p}), \hat{u}_\tau(\mathbf{q}))| \leq \frac{72\epsilon'}{\nu\lambda}$. Before proving our main resut, we need a few more assumptions. Define $\kappa = \max(\log(1/p), \theta)$. We make the simplifying assumption that the number of samples m is sufficiently large such that $\epsilon' \leq \kappa$ and $\phi \geq 12\epsilon'$. This is a reasonable assumption as we typically assume that ϵ' is small. The following lemma forms a bound with the numerator of (18).

Lemma 6.

$$\left|\log\left(\frac{D^{\mathbf{p}}(\tau)D^{\mathbf{q}}(\eta)}{D^{\mathbf{p}}(\eta)D^{\mathbf{q}}(\tau)}\right)\beta(\gamma_\eta, u_\eta(\mathbf{p}), u_\eta(\mathbf{r})) - \log\left(\frac{\hat{D}^{\mathbf{p}}(\tau)\hat{D}^{\mathbf{q}}(\eta)}{\hat{D}^{\mathbf{p}}(\eta)\hat{D}^{\mathbf{q}}(\tau)}\right)\hat{\beta}(\gamma_\eta, \hat{u}_\eta(\mathbf{p}), \hat{u}_\eta(\mathbf{r}))\right| \tag{40}$$

$$\leq \left|\log\left(\frac{D^{\mathbf{p}}(\tau)D^{\mathbf{q}}(\eta)}{D^{\mathbf{p}}(\eta)D^{\mathbf{q}}(\tau)}\right)\right|\frac{72\epsilon'}{\nu\lambda} + \theta\left|\log\left(1+\epsilon'\right)^4\right| + \frac{72\epsilon'}{\nu\lambda}\left|\log\left(1+\epsilon'\right)^4\right| \tag{41}$$

$$\leq \log\left(\frac{1}{\rho}\right)\frac{144\epsilon'}{\nu\lambda} + 4\theta\epsilon' + \frac{288\epsilon'^2}{\nu\lambda} \tag{42}$$

$$\leq \frac{1}{\nu\lambda}\left[\log\left(\frac{1}{\rho}\right)144\epsilon' + 8\theta\epsilon' + 288\epsilon'^2\right] \tag{43}$$

$$\leq 440\frac{\kappa}{\nu\lambda}\epsilon' \tag{44}$$

Since there are 2 such terms in the numerator, the error bounded by twice of that in Lemma 6. The following lemma deals with the denominator.

Lemma 7. *Write x_1, x_2, x_3, x_4 be the four β terms in the denominator. Using Lemma 4,*

$$|\hat{x}_1\hat{x}_2 - x_1x_2| \leq \frac{72\epsilon'}{\nu\lambda}\left(2\theta + \frac{72\epsilon'}{\nu\lambda}\right) \tag{45}$$

This bound holds for $|\hat{x}_3\hat{x}_4 - x_3x_4|$ as well.

$$\left|\frac{1}{x_1x_2 - x_3x_4} - \frac{1}{\hat{x}_1\hat{x}_2 - \hat{x}_3\hat{x}_4}\right| \leq \frac{|\hat{x}_1\hat{x}_2 - x_1x_2| + |\hat{x}_3\hat{x}_4 - x_3x_4|}{|x_1x_2 - x_3x_4|\,|\hat{x}_1\hat{x}_2 - \hat{x}_3\hat{x}_4|} \tag{46}$$

$$\leq \frac{\frac{144\epsilon'}{\nu\lambda}\left(2\theta + \frac{72\epsilon'}{\nu\lambda}\right)}{\phi\,|\hat{x}_1\hat{x}_2 - \hat{x}_3\hat{x}_4|} \tag{47}$$

$$\leq \frac{\frac{144\epsilon'}{\nu\lambda}\left(2\theta + \frac{72\epsilon'}{\nu\lambda}\right)}{\phi\left(\phi - 4\theta\frac{72\epsilon'}{\nu\lambda} - 2\left(\frac{72\epsilon'}{\nu\lambda}\right)^2\right)}, \tag{48}$$

where the assumption of the smallness of ϵ' ensures that the second term in the product of the denominator is sufficiently larger than 0.

Due to time constraints, we were not able to completely derive the full convergence rate. However, we believe that combining the above lemmas appropriately would yield a desired result (whether it is a desirable rate is a separate matter).