

GuessTuples Project

Andrew J. Wren

21 April 2021

Abstract

Notes on GuessTuples project

1 Configuring the nets

1.1 Alice

The input array to guess is $\mathbf{x} = (x_j)_{j=0,\dots,N_{\text{elements}}}$. There should be N_{code} outputs taking values $\mathbf{y} = (y_j)_{j=0,\dots,N_{\text{code}}}$.

Normalise all the rewards so that for each bit j , $r_{j\checkmark} + (N_{\text{code}} - 1) r_{j\text{x}} = 0$. In other words

$$r_{jk} \leftarrow r_{jk} - \frac{r_{j\checkmark} + (N_{\text{code}} - 1) r_{j\text{x}}}{N_{\text{code}}}. \quad (1)$$

The Q estimate is then taken to be

$$Q(\mathbf{x}) = \sum_j b_j y_j \equiv \sum_j |y_j|, \quad (2)$$

where

$$b_j = \text{sgn}(y_j) \quad (3)$$

is the prediction for the machine value of the j th bit. The loss function is

$$L = |Q(\mathbf{x}) - r|^2. \quad (4)$$

1.2 Bob

Bob receives a matrix, $\mathbf{X} = (X_i) = (X_{ij})$ for $0 \leq i < N_{\text{select}}$, $0 \leq j < N_{\text{elements}}$, and a code $\mathbf{c} = (c_k)_{k=0,\dots,N_{\text{code}}}$. Why not makes his outputs be Q -estimates $\mathbf{z} = (z_i)_{i=0,\dots,N_{\text{select}}}$. Bob's prediction is then $\mathbf{x}_{\text{pred}} = \mathbf{X}_{i_{\text{pred}}}$ where

$$i_{\text{pred}} = \text{argmax}_i (z_i). \quad (5)$$

The loss function is

$$L = |z_{i_{\text{pred}}} - r|^2. \quad (6)$$

How do we enforce covariance with respect to the order of (X_i) ?

1. Covariance will occur naturally and quickly without any specific intervention. *To be determined.*
2. Covariance can be enforced through choosing a set $\{\sigma\} \subseteq S_{n_{\text{code}}}$, which could be generated element-by-element by composing randomly-selected basis transpositions $(j \ j + 1)$, and then adding to the loss a term

$$\mu \sum_{\sigma} |z - \sigma^{-1} [z(\sigma[\mathbf{X}])]|^2 \quad (7)$$

for some fixed hyperparameter $\mu > 0$. Note this the term is still run backward through the original $\mathbf{x} \mapsto \mathbf{z}$ net configuration only. *How effective would that be? How big does $\{\sigma\}$ have to be? And how much time would the permutation and the additions forward passes cost?*

3. Enforce covariance via direct identification of weights in Bob's net. *How?*
4. Something related to set transformers. ?
5. Adopt a different basic set-up where each (X_i) is fed through the net separately, alongside the code \mathbf{c} , resulting in a Q -estimate z_i . Then find the loss function as in Eq. 6. *Seems the most straightforward? Adopt this for now.*

None of these quite amount to Bob seeks to reproduce the Alice's code vocabulary. However Bob could additionally set up a net in the same basic configuration as Alice's (he doesn't know the weights of course) and train *that* net jointly with his main net.