1. Abstract
   1. Barcodes in memory allow for search and recall, but what if the barcodes are very close?
   2. In an ideal world, distinctions can always be made, but there cannot be an assumption of a lack of noise being introduced at some point in the process
   3. By transforming the barcodes into a different space via an intermediate model, we are able to show a resilience to noise, while maintaining better performance than previous model attempts (specifically Ritter et al. 2018)
2. Background
   1. Ritter Summary
      1. Based of L2RL (Wang et al. 2016)
      2. Inputs are passed into a single cell LSTM, and gets transformed into a hidden state, which is passed into an A2C policy maker to choose the next action for input into the LSTM
      3. At the end of an episode, a copy of the cell state for the LSTM is stored in a memory buffer, referenced by a key. In our chosen task, this key would be the barcode for that specific episode.
      4. Every input is now also matched against the key in memory, and the prior LSTM cell state found by a 1NN match over the keys is reintegrated via the R-Gate proposed by Ritter.
3. Methods
   1. Task Definition
      1. Contextual Bandits with optional initial clustering of barcodes
      2. Using Hamming distance as a clustering metric, we are able to artificially clump barcodes together and assign/shuffle arm assignments to these clusters over the course of training and evaluation
         1. Intracluster Hamming distance would be less than some chosen threshold
         2. Intercluster Hamming distance would be greater than (barcode\_size – 2\* hamming\_threshold)
         3. Example: 3 arms, 6 distinct barcodes, each barcode is a 16-dimensional binary vector, and we choose a Hamming threshold of 3.
            1. We would create 2 clusters of 3 barcodes (mapping arms 1, 2, 3 to a barcode in each cluster)
            2. The differences between the barcodes of a single cluster would all be at most 3
            3. The difference between the centers of the clusters would be at least (16-2\*3) 10
            4. This ensures even under noise, it is very unlikely for a barcode in cluster 1 to be mistaken for a barcode in cluster 2, but there is high chance for a cluster 1 barcode to be misinterpreted as another cluster1 barcode
            5. The misinterpretation of the barcode would lead to the agent choosing to pull the wrong arm as the next action.
      3. We define an episode as a series of 10 pulls, with rewards generated by the ground truth (unnoised) barcode.
      4. An epoch is a randomized run through all of the N barcodes, with repetition of each barcode N times.
         1. If there are 6 distinct barcodes, we would have 36 episodes, where each individual barcode is presented 6 times.
      5. The mapping of arms to barcode is shuffled at the end of every epoch, with clusters maintaining their unique arm choices
         1. Assume A, B, C were a single cluster, and D, E, F were the other cluster.
         2. If the first epoch had the following mapping: {A:1, B:2, C:3, D:1, E:2, F:3} then a reshuffle would possibly be {A:2, B:3, C:1, D:3, E:1, F:2}
         3. We don’t allow the following reshuffle: {A:1, B:1, C:3, D:2, E:2, F:3} since that makes cluster 1 have 2 instances of arm 1, and cluster 2 have 2 instances of arm 2
   2. Noise Definition
      1. At the beginning of an episode, we choose N bits and randomly decide to flip or keep those bits the same. The rest of the barcode is unchanged. This applied noise is held constant over the course of the episode
      2. We evaluated over 4 different levels of noise, specifically 25%, 50%, 75% and 87.5%
         1. For a 16 dim barcode, this would correspond to random flips on 4, 8, 12, or 14 of the bits of the barcode.
   3. Supervised Learning
4. Results
5. Conclusions
6. Future work