Machine Learning **Fall 2023**

Lecture 13: Multi-Arm Bandits

Lecturer: Liwei Wang *Scribe: Xin Hao, Xiang Wang, Yang Li*

13.1 Definition

The multi-arm Bandits Problem is a simplified reinforcement learning problem. Because we don't know the actual reward, the trade-off between exploration and exploitation is necessary.

Arm 1 2 3 ... k
parameters $\mu(1)$ $\mu(2)$ $\mu(3)$ ... $\mu(k)$, $\mu(i) \in [0,1]$.

1. The loss of an arm a is iid random, with expectation $\mu(a)$, loss$\in [0,1]$.
2. $a^* = argmin_a \mu(a)$.
3. Regret $= \sum_{t=1}^{T} l(a_t) - T\mu(a^*)$.

13.2 Upper Confidence Bound(UCB) algorithm

Key Idea: Optimism in the face of uncertainty. To be exact, after t rounds, each arm has an estimated interval of loss (with high probability), and we just assume the lowest bound of the interval is the loss probability of the arm. The algorithm is called the upper confidence bound because people used to consider the probability of a win in the past. The implementation of the UCB algorithm is as follows. Note

Algorithm 10.2.1: UCB Algorithm

Initialization: $n_0(a) = 0(\forall a \in [k])$;
$n_t(a)$ represents #times arm a is pulled at time t;
$\hat{\mu}_t(a)$ represents the empirical loss of arm a at time t;
for $t = 1, 2, \cdots, T$ **do**
  For each arm a, compute $\mathrm{UCB}_t(a) = \hat{\mu}_{t-1}(a) - \sqrt{\frac{\ln T}{n_{t-1}(a)}}$;
  Pull the arm $a_t = \arg\min_{a \in [k]} \mathrm{UCB}_t(a)$;
  Update $n_t(a_t), \hat{\mu}_t(a_t)$;
end

Figure 13.1: UCB Algorithm

that at the very beginning, if there exists $n_t - 1(a) = 0$, then $UCB_{t(a)} = -\infty$. Therefore the algorithm tends to explore all arms until each arm has been chosen once.

13.2.1 Regret of UCB

Theorem 13.1 *the regret of the UCB algorithm can be bounded as:*

$$R(T) := \mathbb{E}\left[\sum_{t=1}^{T} l(a_t)\right] - T\mu(a^*) \leq \sum_{a:\Delta_a>0} (\frac{16 \ln T}{\Delta_a} + 2\Delta_a)$$

where $\Delta_a = \mu(a) - \mu(a^*)$.

If the gap Δ_a are constants, then Regret$\leq O(K \ln T)$

Notice that when Δ_a is small (e.g. $1/T$), the given upper bound can be very large, which is counter-intuitive. To fix this problem, we revise the bound as

$$\begin{cases} R(T) \leq O(\sqrt{T} \ln T) & , \Delta_a \geq \frac{1}{\sqrt{T}} \\ R(T) \leq \sqrt{T} & , \Delta_a < \frac{1}{\sqrt{T}} \end{cases}$$

So we have $R(T) \leq O(\sqrt{T} \ln T) + \sqrt{T} = O(\sqrt{T} \ln T)$

13.2.2 Thompson Sampling

The Frequency and Bayesian is two school in statistics. The Frequency School take the parameter in the model as a fix value and use MLE to estimate it , while Bayesian School take the parameter as a random variable, and try to induce its posterior distribution.

Thompson Sampling is a Bayesian method that assumes arm a to have Beta distribution, and update the posterior distribution per step to estimate the optimal choice.

13.3 Chain of Thought

Autoregressive Transformers

- Most LLMs follow the autogressive design paradigm.
- Main idea: various tasks can be uniformly treated as sequence generation problems
- The input along with the task description can be together encoded as a sequence of tokens, called the prompt
- The answer is generated by predicting subsequent tokens conditioned on the prompt in an autoregressive way.

Chain of Thought Prompting (CoT)

- Crucial for tasks involving math or reasoning
- Two typical triggering methods:Zero-shot and Few-shot

Questions

1. How can we theoretically understand the power of CoT generation?
2. How can these prompts trigger the CoT generation? Can we design better prompting strategies to further exploit the power of LLMs?
3. How can CoT emerge in LLMs trained over massive data?

and we focus on the first question.

13.3.1 Autoregressive Tansformers

Input : a sequence of tokens s of length n.

Initial embedding : $X^{(0)} = [v_1 + p_1, \ldots, v_n + p_n]^T \in \mathbb{R}^{n \times d}$, where

- each input token s_i is converted to a d-dimensional vector $v_i = Embed(s_i) \in \mathbb{R}^d$
- $p_i \in \mathbb{R}^d$ is the positional embedding.

Propagation: L Transformer blocks follow, each of which transforms the input by

$$X^{(l)} = X^{(l-1)} + Attn^{(l)}(X^{(l-1)}) + FFN^{(l)}\left(X^{(l-1)} + Attn^{(L)}(X^{(l-1)})\right)$$

- $Attn^{(l)}$ is a multi-head self-attention layer,
- $FFN^{(l)}$ is a 2-layer feed forward network with GeLU activation.

$$FFN^{(l)}(X) = \sigma(XW_1^{(l)})W_2^{(l)}$$

13.3.2 Multi-Head self Attention

CoT is The Key to Solving Math Problems

- Problem Formulation
- Log-precision Transformer
- Negative Results
  - Assume $TC^0 \neq NC^1$. For any prime number p, integer L, and any polynomial Q
  - there exists a problem size n such that no log-precision auto regressive Transformer with depth L and hidden dimension $d < Q(n)$ can directly solve the problem Arithmetic(n,p)
  - there exists a problem size m such that no log-precision auto regressive Transformer with depth L and hidden dimension $d < O(m)$ can directly solve the problem Equation(m, p)
- Key Insight:Circuit Complexity
- How about Generating a CoT solution?

  - For any prime p. For any integer n ¿ 0, there exists an auto regressive Transformer with constant hidden size d (independent of a), depth $l = \delta$ and δ heads in each layer that can generate the CoT solution for all inputs in Arithmetic(n, p).Moreover, all parameter values in the Transformer are bounded by $O(poly(n))$.
  - Fix any prime p. For any integer m ¿ 0, there exists an auto regressive Transformer with constant hidden size d (independent of m), depth $L = \delta$, and δ heads in each layer that can generate the CoT solution for all inputs in Equation(rm, p).Moreover, all parameter values in the Transformer are bounded by $O(poly(m))$

- Discussion

CoT is the Key to Solve General Problems

- CoT can Implement Dynamic Progarmming
- Experiments : Length Extrapolation: Long Sequence
- Future Work
  - How do prompt influent the CoT generation?
  - How to understand the role of model size played in CoT generation quailty?
  - How can CoT emerge in LLMS trained over massive data?
  - Designing better architechtures to solve complex without resorting to CoT generation

References

[AGM97] N. Alon, Z. Galil and O. Margalit, On the Exponent of the All Pairs Shortest Path Problem, *Journal of Computer and System Sciences* **54** (1997), pp. 255–262.

[F76] M. L. Fredman, New Bounds on the Complexity of the Shortest Path Problem, *SIAM Journal on Computing* **5** (1976), pp. 83-89.