Name:

**University of Illinois** Spring 2020

# CS 446/ECE 449 Machine Learning
# Homework 10: REINFORCE

Due on Tuesday May 5 2020, noon Central Time

1. **[16 points]** REINFORCE

We are given a utility $U(\theta) = \mathbb{E}_{p_\theta}[R(y)] = \sum_{y\in\mathcal{Y}} p_\theta(y)R(y)$ which is the expected value of the non-differentiable reward $R(y)$ defined over a discrete domain $y \in \mathcal{Y} = \{1, \ldots, |\mathcal{Y}|\}$. Our goal is to learn the parameters $\theta$ of a probability distribution $p_\theta(y)$ so as to obtain a high utility (high expected reward), *i.e.*, we want to find $\theta^* = \arg\max_\theta U(\theta)$. To this end we define the probability distribution to read

$$p_\theta(y) = \frac{\exp F_\theta(y)}{\sum_{\hat{y}\in\mathcal{Y}} \exp F_\theta(\hat{y})}. \quad (1)$$

(a) (3 points) If we are given an i.i.d. dataset $\mathcal{D} = \{(y)\}$ we can learn the parameters $\theta$ of a distribution via maximum likelihood, *i.e.*, by addressing

$$\max_\theta \sum_{y\in\mathcal{D}} \log p_\theta(y)$$

via gradient descent. State the cost function and its gradient when plugging the model specified in Eq. (1) into this program. When is this gradient zero?

**Solution:** Cost function:

$$\sum_{y\in\mathcal{D}} \left[ F_\theta(y) - \log \sum_{\hat{y}\in\mathcal{Y}} \exp F_\theta(\hat{y}) \right]$$

Gradient:

$$\sum_{y\in\mathcal{D},\hat{y}\in\mathcal{Y}} [\delta(y=\hat{y}) - p_\theta(\hat{y})] \frac{\partial F_\theta(\hat{y})}{\partial\theta} \qquad \text{where} \quad \delta(y=\hat{y}) = \begin{cases} 1 & \text{if } \; y = \hat{y} \\ 0 & \text{otherwise} \end{cases}$$

Gradient zero if prediction equals groundtruth

(b) (2 points) If we aren't given a dataset but if we are instead given a reward function $R(y)$ we search for the parameters $\theta$ by maximizing the utility $U(\theta)$, *i.e.*, the expected reward. Explain how we can approximate the utility by sampling from the probability distribution $p_\theta(y)$.

**Solution:**
Draw samples $\tilde{y}_i \sim p_\theta(y)$ and average their rewards, *i.e.*, we approximate:

$$U(\theta) = \mathbb{E}_{p_\theta}[R(y)] = \sum_{y\in\mathcal{Y}} p_\theta(y)R(y) \approx \frac{1}{N}\sum_{i=1}^{N} R(\tilde{y}_i) \qquad \text{where} \quad \tilde{y}_i \sim p_\theta(y)$$

(c) (3 points) Using general notation, what is the gradient of the utility $U(\theta)$ w.r.t. $\theta$, *i.e.*, what is $\nabla_\theta U(\theta)$. How can we approximate this value by sampling from $p_\theta(y)$? Make sure that you stated the gradient in the form which ensures that computation via sampling from $p_\theta(y)$ is possible.

**Solution:**
Derivation in class slides

$$\nabla_\theta U(\theta) = \mathbb{E}_{p_\theta}[R(y)\nabla_\theta \log p_\theta(y)] = \sum_{y\in\mathcal{Y}} p_\theta(y)R(y)\nabla_\theta \log p_\theta(y) \approx \frac{1}{N}\sum_{i=1}^{N} R(\tilde{y}_i)\nabla_\theta \log p_\theta(\tilde{y}_i)$$

where

$$\tilde{y}_i \sim p_\theta(y)$$

(d) (5 points) Using the parametric probability distribution defined in Eq. (1), what is the approximated gradient of the utility? How is this gradient related to the result obtained in part (a)?

**Solution:**

$$\frac{1}{N}\sum_{i=1}^{N} R(\tilde{y}_i)\left(\sum_{\hat{y}\in\mathcal{Y}}[\delta(\tilde{y}_i = \hat{y}) - p_\theta(\hat{y})]\frac{\partial F_\theta(\hat{y})}{\partial\theta}\right) \qquad \text{where} \quad \tilde{y}_i \sim p_\theta(y)$$

We no longer have a groundtruth signal that tells us in what direction to change the prediction. Instead we use the reward $R(\tilde{y}_i)$ to emphasize directions which give large rewards.

(e) (3 points) In A10_Reinforce.py we compare the two forms of learning. Let the size of the domain $|\mathcal{Y}| = 6$, and let the groundtruth data distribution $p_{\mathrm{GT}}(y) = 1/12$ for $y \in \{1, 6\}$, $p_{\mathrm{GT}}(y) = 2/12$ for $y \in \{2, 5\}$, and $p_{\mathrm{GT}}(y) = 3/12$ for $y = \{3, 4\}$. The dataset $\mathcal{D}$ contains $|\mathcal{D}| = 1000$ points sampled from this distribution. Further let $F_\theta(y) = [\theta]_y$, where $\theta \in \mathbb{R}^6$ and where $[a]_y$ returns the $y$-th entry of vector $a$. The reward function happens to equal the groundtruth distribution, *i.e.*, $R(y) = p_{\mathrm{GT}}(y)$. What distribution $p_\theta$ is learned with the maximum likelihood approach? What distribution is learned with the REINFORCE approach? Explain why this is expected. Complete A10_Reinforce.py to answer these questions.

**Solution:**
Maximum likelihood: since the dataset size $|\mathcal{D}|$ is reasonably large compared to the domain size $|\mathcal{Y}|$ an accurate distribution can be learned, *i.e.*,

$$p_{\mathrm{GT}}(y) \approx p_{\theta_{\mathrm{ML}}}(y)$$

REINFORCE: REINFORCE attempts to learn a distribution which maximizes the utility when sampled from it, hence we expect

$$p_{\theta_{\mathrm{R}}}(3) = 1 \qquad \text{or} \qquad p_{\theta_{\mathrm{R}}}(4) = 1$$

and all other entries to equal zero. Likely only one of the entires will equal one due to sampling inbalance.