EN.601.774 Theory of Replicable ML Spring 2025

Lecture 4

Instructor: *Jess Sorrell* Scribe: *Jess Sorrell*

Statistical Queries Cont.

Last time we introduced statistical queries and the PAC learning framework. We showed that we could express an ERM learner for finite hypothesis classes as a non-adaptive SQ algorithm, and therefore PAC-learn finite hypothesis classes with $m \in O(\frac{\log(|\mathcal{H}|/\delta)}{\varepsilon^2})$ samples.

Definition 0.1 (Statistical Queries [Kearns, 1998])**.** Let $\mathcal{X}$ denote a domain. A *statistical query* is a function of the form $\phi : \mathcal{X} \to [0, 1]$. Let D be a distribution on $\mathcal{X}$. The *value* of a statistical query ϕ on D is defined $\mathbb{E}_{x\sim D}[\phi(x)]$ (abbreviated $\mathbb{E}_D[\phi]$). We will similarly use the abbreviation $\mathbb{E}_S[\phi] = \frac{1}{m}\sum_{x\in S}\phi(x)$.

We said that the some popular algorithms in ML can be expressed as statistical query algorithms.

- gradient descent
- Markov chain monte carlo
- PCA
- K-means clustering

To see that gradient descent can be formulated as an SQ algorithm, we first define the model parameter space $\Theta \in \mathbb{R}^d$ and an L-Lipschitz loss $\ell(\theta, x)$. Then the gradient we're interested in computing is $\nabla \ell_D(\theta) = \langle \frac{\partial \ell_D(\theta)}{\partial \theta_1}, \ldots, \frac{\partial \ell_D(\theta)}{\partial \theta_d} \rangle$. To express $\frac{\partial \ell_D(\Theta)}{\partial \Theta_i}$ as a statistical query, we take $\phi_{\theta,i}(x, y) = \frac{1}{L}\frac{\partial \ell_{(x,y)}(\theta)}{\partial \theta_i}$ (scaling to ensure $\phi_{\theta,i} \to [0, 1]$). Then $a_{\theta,i} \in \mathbb{E}_D[\frac{\partial \ell_D(\theta)}{\partial \theta_i} \pm \tau]$. Estimating these d queries to within τ ensures we estimate the true gradient to within ℓ_2 error $\sqrt{d}L\tau$.

To see that K-means clustering can be formulated as an SQ algorithm, let's recall the iterative refinement procedure of Lloy'd algorithm. We begin with a guess at the k means $m_1, \ldots, m_k$, then:

1. We partition the dataset into k clusters, where

$$S_i = \{x \in S : \|x - m_i\| \leq \|x - m_j\| \forall j \in [k], j \neq i\}$$

2. $\forall i \in [k]$, we update $m_i = \frac{1}{|S_i|}\sum_{x\in S_i} x$

To express this as an SQ algorithm, we'll need two different kinds of queries.

$$\phi_{|S_i|}(x) = \frac{1}{|S|} \cdot \begin{cases} 1, & \|x - m_i\| \leq \|x - m_j\| \forall j \in [k], j \neq i \\ 0, & o.w. \end{cases}$$

Then $a_{|S_i|} = \mathbb{E}_S[\phi_{|S_i|}(x)] = \frac{|S_i|}{|S|}$. We'll also need to compute each component of the centroid of each partition element. For all $i \in [k]$, $j \in [d]$,

$$\phi_{m_i,j}(x) = \begin{cases} x_j, & \|x - m_i\| \leq \|x - m_j\| \forall j \in [k], j \neq i \\ 0, & o.w. \end{cases}$$

Then $a_{m_i,j} = \mathbb{E}_S[\phi_{m_i,j}(x)] = \frac{1}{|S|}\left(\sum_{x \in S_i} x_j + \sum_{x \notin S_i} 0\right)$. So

$$\frac{a_{m_i m j}}{a_{|S_i|}} = \frac{\frac{1}{|S|}\left(\sum_{x \in S_i} x_j + \sum_{x \notin S_i} 0\right)}{\frac{|S_i|}{|S|}} = \frac{1}{|S_i|} \sum_{x_j \in S_i} x$$

So each update to each of the k m_i's can be computed with $d + 1$ SQs.

References

Michael Kearns. Efficient noise-tolerant learning from statistical queries. *Journal of the ACM (JACM)*, 45(6):983–1006, 1998.