Project Report - CS773A
Mixed Linear Regression in Online Setting

Avinash Mohak, Sarthak Garg and Utkarsh Agarwal

1 Introduction

We consider the MLR problem in a two component setting i.e. where each sample comes from exactly one of the two unknown vectors but the exact identity of the source is unknown. Consider a set of N measurements $(y_i, \mathbf{x}_i)$ i.e. for $i = 1, ..., N$

$$y_i = \langle \mathbf{x}_i, \beta_1^* \rangle z_i + \langle \mathbf{x}_i, \beta_2^* \rangle (1 - z_i) + \eta_i$$

where z_i is either 0 or 1, depending on whether the i^{th} observation comes from β_2^* or β_1^* respectively and η_i is noise. Our aim is to estimate $\beta_1^*, \beta_2^* \in \mathbb{R}^k$ in an online setting given only the measurements $(y_i, \mathbf{x}_i)$, $i = 1, ..., N$.

The problem has quite a few interesting applications. One of them is in e-commerce marketplace. As it often happens we tend to search and purchase products on e-commerce sites for our friends, family and who knows who from a single user account. The recommendation system often goes berserk as it tries to fit a single model for such a varied range of searches but little does it know, that behind the scene there are multiple users disguised as one. Since the searches and prediction happen sequentially, this makes a classic case for the problem setting.

MLR has seen quite a few developments in the offline setting which we discuss below but it is still yet to be explored in the online setting. We aim to explore this area and to develop an algorithm for the online setting.

2 Prior Work

Most of the recent work on this problem has been done for the batch setting.

2.1 Alternating Minimization in Mixed Linear Regression

[**?**]The authors used the classical EM algorithm for estimating the model vectors. They showed that with a suitable initialization scheme, the EM algorithm provably converges. The results obtained were conditioned on the assumptions that the context vectors $\mathbf{x}_i$ are sampled i.i.d from the normal distribution and that there is no noise in the models. The proposed initialization algorithm returns β_1^0 and β_2^0 which are within a constant distance from the true vectors β_1^* and β_2^* with high probability given that the sample size is $O(k \log^2(k))$ where k is the dimension of the context vectors. The authors further show that given this guarantee on the initial guesses, each step of the EM algorithm reduces the error by $\frac{1}{2}$ implying geometric convergence.

2.2 Mixed Linear Regression with Multiple Components

[**?**] The authors gave the first exact recovery guarantees for the case when the MLR model has more than 2 components. They propose the following loss function for the problem

$$\ell(\mathbf{w}_1, \mathbf{w}_2..\mathbf{w}_K) = \sum_{i=1}^{K} \prod_{k=1}^{K} (y_i - < \mathbf{x}_i, \mathbf{w}_k >)^2$$

Here $\mathbf{w}_i$ are the model vectors. This loss function is then shown to be locally strongly convex in the neighborhood of the true model vectors. The authors use a suitable initialization scheme to confine the initial guesses in this neighborhood. Then they use convex optimization algorithms to minimize the loss function. Let N be the number of samples, d be the dimension of the context vectors and K be the number of underlying models. The proposed algorithm has a computational complexity of $O(Nd)$ and a sample complexity of $O(d(K\log(d))^K)$

2.3 A Convex Formulation for Mixed Regression with Two Components

[?] The authors consider the case when there are 2 underlying models with the inclusion of stochastic and adversarial noise. They reduce the problem to a convex optimization problem, which recovers the exact model vectors. Let d be the dimension of the context vectors and $\mathbf{e}$ be the vector of adversarial errors in the observations and n be the total number of samples. The authors showed that if the number of observations for each regressor exceed d, then their algorithm recovers estimates $(\hat{\beta}_1, \hat{\beta}_2)$ such that

$$||\hat{\beta}_b - \beta_b^*|| \leq \frac{||\mathbf{e}||_2}{\sqrt{n}}, b = 1, 2.$$

3 Notation and Setup

We will solve the problem in a no noise setting i.e. $\eta_i = 0$. As shown in [?], the problem setting is NP hard in offline setting without any further assumptions. As in [?], we would also carry forward with the assumption that the measurement vectors $\mathbf{x}_i$ are uniform and independent Gaussian vectors in $\mathbb{R}^d$.
Let T be the number of rounds and $D = \{0, 1\}$ be the set of decisions. In each round t, we receive a measurement $(y_i, \mathbf{x}_i)$ and we must make a decision $z^t \in D$. Each such decision results in a loss ℓ^t. We then update our estimates β_1^t and β_2^t based on ℓ^t to get β_1^{t+1} and β_2^{t+1}respectively. Also define $err^t = \max\{\|\beta_1^t - \beta_1^*\|_2, \|\beta_2^t - \beta_2^*\|_2\}$ which is a measure of how far are the predictions from the true estimate.

4 Algorithm

The following algorithm states a generic version for solving mixed linear regression. Since the true label z_t corresponding to the origin of the context is unknown even to the adversary, we have to make do with the loss $\ell(\hat{y}^t, y^t)$.

Algorithm 1 MLR in Online Setting

```
for $t = 1, ......, T$ do
    Receive context $\mathbf{x}^t$
    Make prediction $\hat{y}^t \in D$
    Receive true value, $y^t$
    Incur loss $\ell^t = \ell(\hat{y}^t, y^t)$
    $\beta_1^{t+1} \longleftarrow$ Update $\beta_1^t$
    $\beta_2^{t+1} \longleftarrow$ Update $\beta_2^t$
end for
return $\beta_1^T$ and $\beta_2^T$
```

We now present our proposed algorithm. The algorithm is split into two parts. One part is the initialisation which initialises our estimates β_1^t and β_2^t to bring the estimates in close proximity of β_1^* and β_2^* respectively. The other part applies standard ordinary least squares regression problem by splitting the points available till now $(x^1, \ldots, x^t)$ on the basis of our estimates β_1^t and β_2^t and utilising the split to recover better estimates β_1^{t+1} and β_2^{t+1}. But the issue here is that to obtain a good initialisation we need $O(klog^2k)$ samples, thus only after $O(klog^2k)$ iterations we can hope to start making sensible predictions for the points x_t and only then our updates make sense.

Algorithm 2 Initialisation with Resampling

```
**Input** Grid Resolution $\delta$, samples$\{(y_i, \mathbf{x}_i, i = 1, \ldots, N\}$
Partition the samples into two disjoint sets: $S_*, S_+$
$M \leftarrow \sum_{i \in S^*} y_i^2 \mathbf{x}_i \otimes \mathbf{x}_i$
Compute top two eigen vectors $\mathbf{v}_1, \mathbf{v}_2$ of $M$
*Make the grid points*: $G \leftarrow \{\mathbf{u} : \mathbf{u} = \mathbf{v}_1 cos(\delta t) + \mathbf{v}_2 sin(\delta t), t = 0, 1, \ldots, \lceil \frac{2\pi}{\delta} \rceil\}$
$\beta_1^{t,0}, \beta_2^{t,0} \leftarrow \underset{\mathbf{u}_1, \mathbf{u}_2 \in G}{\text{argmin}} L_{S_+}(\mathbf{u}_1, \mathbf{u}_2)$
**return** $\beta_1^{t,0}$ and $\beta_2^{t,0}$
```

Note that $L_S(\beta_1, \beta_2)$ is the least squared loss incurred on the sample set S when it partitioned on the basis of the proximity to β_1 or β_2.

Algorithm 3 EM with Resampling

```
**Input** Initial $\beta_1^{t,0}, \beta_2^{t,0}$, samples$\{(y_i, \mathbf{x}_i, i = 1, \ldots, N\}$
$J_1, J_2 \leftarrow \phi$
**for** $i = 1, ......, N$ **do**
    **if** $|y_i - \langle \mathbf{x}_i, \beta_1^{t,0} \rangle| < |y_i - \langle \mathbf{x}_i, \beta_2^{t,0} \rangle|\rangle$ **then**
        $J_1 \leftarrow J_1 \cup \{i\}$
    **else**
        $J_2 \leftarrow J_2 \cup \{i\}$
    **end if**
**end for**
$\beta_1^{t,1} \leftarrow \text{argmin}_\beta \|\mathbf{y}_{J_1} - \mathbf{X}_{J_1}\beta\|$
$\beta_2^{t,1} \leftarrow \text{argmin}_\beta \|\mathbf{y}_{J_1} - \mathbf{X}_{J_2}\beta\|$
**return** $\beta_1^{t,1}$ and $\beta_2^{t,1}$
```

Algorithm 4 Alternating Minimization in Online Setting

```
$J \leftarrow \phi$
**for** $t = 1, ......, T$ **do**
    Receive $\mathbf{x}_t$
    $J \leftarrow J \cup \{\mathbf{x}_t\}$
    **if** $|y_i - \langle \mathbf{x}_i, \beta_1^t \rangle| < |y_i - \langle \mathbf{x}_i, \beta_2^t \rangle|\rangle$ **then**
        Predict $\langle \mathbf{x}_i, \beta_1^t \rangle$
    **else**
        Predict $\langle \mathbf{x}_i, \beta_2^t \rangle$
    **end if**
    **if** $t < C$ **then**
        $\beta_1^{t+1}, \beta_2^{t+1} \leftarrow$ Initialisation with Resampling$(\delta, J)$
    **else**
        Gather points and make mini batch $J_k$ of size $k$
        **if** (Size$(J_k) = k$) **then**
            $\beta_1^{t+1,1}, \beta_2^{t+1,1} \leftarrow$ EM with re-sampling$(\beta_1^{t,1}, \beta_2^{t,1}, J_k)$
        **else**
            $\beta_1^{t+1,1}, \beta_2^{t+1,1} \leftarrow \beta_1^{t,1}, \beta_2^{t,1}$
        **end if**
    **end if**
**end for**
```

By results of [1] the $O(k)$ sample size of updation of β_1, β_2 gives a geometric decrease guarantee in the error with a probability of $1 - exp(-c_2 k)$ i.e. $err^t \leq err^{t-1}$ with probability $1 - exp(-c_2 k)$ and also $C = O(klog^2 k)$ for a good initial estimate.

5 Regret Analysis

We have assumed that the vectors x^t are unit norm and the true vectors β_1^* and β_2^* are also unit norm. These assumptions have also been made in [1] and thus pose no additional restrictions.

$$y_i^t = \langle \mathbf{x}_i^t, \beta_1^* \rangle z_i + \langle \mathbf{x}_i^t, \beta_2^* \rangle (1 - z_i)$$

Now

$$E[y_i^t] = p\langle \mathbf{x}_i^t, \beta_1^* \rangle + (1-p)\langle \mathbf{x}_i^t, \beta_2^* \rangle$$

where p is the proportion of points coming from the first sample.
Since the vectors x_t are being generated stochastically

$$\begin{aligned} l_t &\geq \min\{p(\langle \mathbf{x}_i^t, \beta_1^* \rangle - \langle \mathbf{x}_i^t, \beta_2^* \rangle)^2, (1-p)(\langle \mathbf{x}_i^t, \beta_1^* \rangle - \langle \mathbf{x}_i^t, \beta_2^* \rangle)^2\} \\ &= \min\{p, 1-p\}(\langle \mathbf{x}_i^t, \beta_2^* \rangle - \langle \mathbf{x}_i^t, \beta_1^* \rangle)^2 \end{aligned}$$

where loss is taken to be the squared loss.
Let us denote the instantaneous regret i.e. regret in the t^{th} iteration as r_t. Also, let us assume without loss of generality that $p > \frac{1}{2}$.
So the instantaneous loss incurred by the benchmark B is
$l_t^B = (1-p)[\langle \mathbf{x}^t, \beta_2^* \rangle - \langle \mathbf{x}^t, \beta_1^* \rangle]^2$
Also for now assume that after sufficient number of iterations our estimate $\hat{p} > \frac{1}{2}$ if $p > \frac{1}{2}$*(We will prove this later)*. So

$$\begin{aligned} E[r_t] &= E[l_t] - l_t^B \\ &= pl(\langle \mathbf{x}^t, \hat{\beta}_1 \rangle, \langle \mathbf{x}^t, \beta_1^* \rangle) + (1-p)l(\langle \mathbf{x}^t, \hat{\beta}_1 \rangle, \langle \mathbf{x}^t, \beta_2^* \rangle) - (1-p)[\langle \mathbf{x}^t, \beta_2^* \rangle - \langle \mathbf{x}^t, \beta_1^* \rangle]^2 \\ &= p[\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle]^2 + (1-p)[\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_2^* \rangle]^2 - (1-p)[\langle \mathbf{x}^t, \beta_2^* - \beta_1^* \rangle]^2 \\ &= p[\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle]^2 + (1-p)[\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_2^* - \beta_2^* + \beta_1^* \rangle \langle \mathbf{x}^t, \hat{\beta}_1 - \beta_2^* + \beta_2^* - \beta_1^* \rangle] \\ &= p[\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle]^2 + (1-p)[\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* + 2\beta_1^* - 2\beta_2^* \rangle \langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle] \\ &\leq p\|\hat{\beta}_1 - \beta_1^*\|_2^2 + (1-p)\|\hat{\beta}_1 - \beta_1^*\|^2 + 2(1-p)[\langle \mathbf{x}^t, \beta_1^* - \beta_2^* \rangle \langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle] \\ &\leq \|\hat{\beta}_1 - \beta_1^*\|_2^2 + 2(1-p)[\langle \mathbf{x}^t, \beta_1^* - \beta_2^* \rangle \langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle] \\ &\leq \|\hat{\beta}_1 - \beta_1^*\|_2^2 + \|\beta_1^* - \beta_2^*\|_2 \|\hat{\beta}_1 - \beta_1^*\|_2 \\ &\leq \|\hat{\beta}_1 - \beta_1^*\|_2^2 + 2\|\hat{\beta}_1 - \beta_1^*\|_2 \end{aligned}$$

By the results of the [1] $\|\hat{\beta}_1 - \beta_1^*\|$ reduces geometrically, thus incurring a constant pseudo regret.

5.1 Estimating p

The point x_t is correctly assigned if

$$\begin{aligned} &[\langle \mathbf{x}^t, \hat{\beta}_1 \rangle - \langle \mathbf{x}^t, \beta_1^* \rangle]^2 < [\langle \mathbf{x}^t, \hat{\beta}_2 \rangle - \langle \mathbf{x}^t, \beta_1^* \rangle]^2 \\ &\Rightarrow [\langle \mathbf{x}^t, \hat{\beta}_1 - \beta_1^* \rangle]^2 < [\langle \mathbf{x}^t, \hat{\beta}_2 - \beta_1^* \rangle]^2 \end{aligned}$$

By Result 5.1 of [1], the above holds with probability $q > \frac{1}{2}$ if

$$\|\hat{\beta}_1 - \beta_1^*\| < \|\hat{\beta}_2 - \beta_1^*\|$$

which holds true after sufficient number of iterations by Theorem 4.4 in [1].
Thus our estimate $\hat{p}$ is the fraction of the points classified as originating from the first Gaussian

$$\begin{aligned} \hat{p} &= p(\text{Probability that the point coming from the first Gaussian is classified correctly}) \\ &+ (1-p)(\text{Probability that the point coming from the first gaussian is classified incorrectly}) \\ &= pq + (1-p)(1-q) = 2pq + 1 - p - q \end{aligned}$$

Case 1: $p > \frac{1}{2}$

$$\begin{aligned} & p - \hat{p} = 2p + q - 2pq - 1 = (1-q)(2p-1) \leq \frac{1}{2}(2p-1) = p - \frac{1}{2} \\ \Rightarrow & \hat{p} > \frac{1}{2} \end{aligned}$$

Case 2: $p < \frac{1}{2}$

$$\begin{aligned} & \hat{p} - p = 2pq + 1 - 2p - q = (1-q)(1-2p) \leq \frac{1}{2}(1-2p) = \frac{1}{2} - p \\ \Rightarrow & \hat{p} < \frac{1}{2} \end{aligned}$$

Thus after sufficient number of iterations, we get a suitable estimate of p which enables us to incur constant regret from there on.

6 Future Work

6.1 Improvement in our current work

In the above analysis, we have assumed the setting, i.e. the losses to be derived from a no noise setting, which gave us constant pseudo-regret. It would be interesting to analyze the problem in noisy setting, where the noise may enjoy some properties, which can be justified. In addition to that, our assumption that the vectors come from a uniform Gaussian distribution could be relaxed, and analyzed.

6.2 Bandit Formulation

Alternatively, we also propose a bandit formulation to the mixed linear regression problem described above. Intuitively, at every step, we randomly sample a set of vectors from the uniform Gaussian distribution in $\mathbb{R}^d$. We then consider the sampled vectors as arms to be played, and thus run Bandit Optimization on the set, to propose a vector. We assume the reward pertaining to x^t to be the negative of the loss incurred by it. In this sense, reward maximization boils downs to minimization of the loss function, at every step. Doing so, we maintain a confidence set for both the parameters, and with high probability, one can show that the true estimates will lie in the set. Moreover, the confidence set shrinks after every time step. Thus, after many iterations, the true parameters can be obtained correctly whp.

Algorithm 5 Bandit formulation for Mixed Linear Regression

```
for $t = 1, ......, T$ do
    Randomly sample a set of $C$ vectors from uniform Gaussian distribution, say, $\{x_1^t, x_2^t, \cdots x_C^t\}$
    Run Lin-UCB/Thompson Sampling for the sampled set of vectors
    Propose an arm (vector) $x_i^t$ based on the above algorithm
    Incur loss and update the current estimates
```

The next step would be to define the notion of regret(pseudo-regret) for this formulation and its analysis.

References

[1] Yi, Xinyang, Constantine Caramanis, and Sujay Sanghavi. ”Alternating Minimization for Mixed Linear Regression.” ICML. 2014.

[2] Abbasi-Yadkori, Yasin, Dávid Pál, and Csaba Szepesvári. ”Improved algorithms for linear stochastic bandits.” Advances in Neural Information Processing Systems. 2011.

[3] Li, Lihong, et al. ”A contextual-bandit approach to personalized news article recommendation.” Proceedings of the 19th international conference on World wide web. ACM, 2010.