Online Active Learning via Thresholding

Paper Review by Hilda Ibriga

Carlos Riquelme   Ramesh Johari   Boasen Zhang

July 19, 2016

Outline

- **Active learning** is the process in which unlabeled instances are dynamically selected for expert labelling, and then a classifier is trained on the labeled data
- In a regression setting, the unlabelled instances are the covariates observations, the labels is the response output y and the classifier is the function relating the covariates to the output y.

- **Data Quality over Quantity** : Good data can be more effective than more data.
- **Reducing data collection cost**
  - Labeling data or running experiments can be tedious, time consuming and expensive.
  - Example : In speech recognition one minute of speech can take ten minutes to label.
- **Budget Constraint**: Learning with the least amount of selected observations.

The Online Marketing Problem

- **Scenario**: A marketing organization plans to send advertisement promotion to a new target market. Their goal is to estimate the revenue that can be expected for individuals with a given covariate vector.
- **Constraint**: Providing the promotion and collecting data on the individual is expensive; they can only afford to advertise to k customers.
- **Naive Solution**: Randomly select k observations units (customers) out of the pool of customers in order to build the predictive model.
- **Better Solution**: An active learning strategy is to select the observations units (customers) which provide the most "information" to the model fitting procedure. Also this is done in an online fashion as opportunities to reach customers arrive sequentially over time.

- Observe iid $X^i \in \mathbb{R}$ sequentially for $i = 1, ..., n$.
- For each X^i choose to observe (label) the output $Y \in \mathbb{R}$ or not.
- Budget: Label (observe) at most k out of the n observations.

Assumptions:

- $X^i \sim D$ is known, $E(X^i) = 0$, $\Sigma = E(XX^T)$ is known.
- $Y = X^T\beta + \varepsilon$, $\beta \in \mathbb{R}^d$, $\varepsilon \sim N(0, \sigma^2)$

Goal: Minimize the expected MSE of the OLS BLUE estimator $\hat{\beta}_k$.

$$\begin{aligned} E(MSE_{\hat{\beta}_k}) &= E[(Y - X^T\hat{\beta}_k)^2] \\ &= E\|\hat{\beta}_k - \beta\|^2 + \sigma^2 \\ &= \sigma^2 E[Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1})] + \sigma^2. \end{aligned}$$

Thresholding Algorithm

Intuition

Assume each X^i is white i.e $E(XX^T) = I$. If not use $X = D^{1/2}U^TX'$ to whiten where $\Sigma = UDU^T$.

- Minimizing the expected MSE is the same as minimizing

$$E[Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1})].$$

We have

$$\frac{d}{\lambda_{max}(\mathbf{X}^T\mathbf{X})} \leq Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \leq \frac{d}{\lambda_{min}(\mathbf{X}^T\mathbf{X})}. \tag{1}$$

- Maximizing $\lambda_{min}(\mathbf{X}^T\mathbf{X})$ the smallest eigenvalue of $\mathbf{X}^T\mathbf{X}$ will minimize the expected MSE .

Thresholding Algorithm

Intuition

Observe that:

- The sum of eigenvalues of $\mathbf{X}^T\mathbf{X} = Tr(\mathbf{X}^T\mathbf{X}) = \sum_{i=1}^{n} \|X^i\|^2$ (the sum of the norms of the observations).

To maximize the smallest eigenvalue $\lambda_{min}(\mathbf{X}^T\mathbf{X})$:

- **Condition 1**: Select observations X^i with large norm.
- **Condition 2**: Select observations so that the eigenvalues of $\mathbf{X}^T\mathbf{X}$ are similar in magnitude.

- Let $\xi \in \mathbb{R}^d$, be a vector of weights with $\sum_{j=1}^{d} \xi_j = d$ and define the norm $\|X\|_{\xi}^2 = \sum_{j=1}^{d} \xi_j X_j^2$.
- Let Γ be a threshold.

Then we have,

$$\lambda_{min}(E\mathbf{X}^T\mathbf{X}) = kmin_j\phi_j \text{ and } \lambda_{max}(E\mathbf{X}^T\mathbf{X}) = kmax_j\phi_j$$

.

Where

$$\phi_j := E_D[X_j^2 | \|X\|_{\xi} \geq \Gamma], \tag{2}$$

are the diagonal elements of the covariance matrix $E_{\bar{D}}(X^i {X^i}^T)$.

- **Condition 1** is achieved by selecting the observations with $\|X\|_\xi \geq \Gamma$ and

$$P_D(\|X\|_\xi \geq \Gamma) = \frac{k}{n}. \tag{3}$$

 The selected observations are iid with distribution $\bar{D} := D$ conditional on $\|X\|_\xi \geq \Gamma$.

- **Condition 2** is achieved by finding $(\xi, \Gamma) \in \mathbb{R}^{d+1}$ such that $min_j \phi_j \approx max_j \phi_j$. That is when

$$E_D[X_j^2 | \|X\|_\xi \geq \Gamma] = \phi_j = \phi \text{ for all } j. \tag{4}$$

Algorithm

- **Step 1**: For each incoming observation X^i, perform whitening and compute its weighted norm $\|X\|_\xi$.
- **Step 2**: If the norm is above Γ, select the observation, otherwise ignore it.
- **Step 3**: Stop when k observations have been collected.

Note: Random sampling is equivalent to $\Gamma = 0$.

Algorithm 1 Thresholding Algorithm.

Set $(\xi, \Gamma) \in \mathbf{R}^{d+1}$ satisfying (3) and (4).
Set $S = \emptyset$.
for observation $1 \leq i \leq n$ **do**
    Observe X^i.
    Compute $X^i = D^{-1/2} U^T X^i$.
    if $\|X^i\|_\xi > \Gamma$ or $k - |S| = n - i + 1$ **then**
        Choose X^i: $S = S \cup X^i$.
        if $|S| = k$ **then**
            break.
        end if
    end if
end for

Adaptive Thresolding

The algorithm can be made adaptive by updating the parameters (ξ_i, Γ_i) after each observation. This is done by finding (ξ_i, Γ_i) such that:

$$P_D(\|X^i\|_{\xi_i} \geq \Gamma_i) = \frac{k - |S_{i-1}|}{n - i + 1}. \tag{5}$$

The adaptive algorithm tends to outperform the simple thresolding algorithm.

Algorithm 1b Adaptive Thresholding Algorithm.

```
Set S = ∅.
for observation 1 ≤ i ≤ n do
    Observe X^i, estimate Σ̂_i = Û_i D̂_i Û_i^T.
    Compute X^i = D̂_i^{-1/2} Û_i^T X_i.
    Let (ξ_i, Γ_i) satisfy (3) and (5).
    if ‖X^i‖_{ξ_i} > Γ_i or k − |S| = n − i + 1 then
        Choose X^i: S = S ∪ X^i.
        if |S| = k then
            break.
        end if
    end if
end for
```

Theoretical Results: Upper Bound

Theorem

Assumptions:

- $d \geq 3$ *and* $n > k > d$.
- $X \in \mathbb{R}^d$ *are distributed according to* D *known and continuous with covariance matrix* $\Sigma \in \mathbb{R}^{d \times d}$.
- X *has fourth moment and marginal density symmetric around zero after whitening.*

Let $\mathbf{X}$ *be a* $k \times d$ *matrix with* k *observations sampled from the distribution induced by the thresholding rule with parameters* $(\xi, \Gamma) \in \mathbb{R}_+^{d+1}$ *satistying* (3). *Let* $\psi \in (0,1)$. *Then there exists a constant* $C_1 > 0$ *(which depends on* D,d,k, n*) such that:*

$$Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \leq \frac{d}{(1-\psi)\phi k} \tag{6}$$

with probability at least $1 - d\exp(-kC_1)$.

Theoretical Results

Corollary

If the observations in Theorem 1 are jointly Gaussian with covariance matrix $\Sigma \in \mathbb{R}^{d \times d}$, *and if,*

- $\xi_j = 1$ *for all* $j = 1, .., d$
- $\Gamma = C\sqrt{d + 2log(n/k)}$ *for some constant* $C \geq 1$,

then,

$$Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \leq \frac{d}{(1-\psi)(1+\frac{2log(n/k)}{d})k} \tag{7}$$

with probability at least $1 - dexp(-kC_1)$.

Gains and Limitations

Gains:

- The MSE of random sampling for white Gaussian data is $\sigma^2 d/k$.
- Active learning provides a gain factor of order $1 + \frac{2log(n/k)}{d}$ with high probability.
- The variance of MSE for a fixed X depends on $\sum_j \frac{1}{\lambda_j(\mathbf{X}^T\mathbf{X})^2}$.
- Active learning therefore decreases the variance of MSE.

Limitations:

- The algorithm suffers from the curse of dimensionality.
- The probability in **upper bound** theorem decreases as d the dimension increases.

Sparse Thresholding Algorithm

Assume $D = N(0, \Sigma)$,
For high dimensional settings ($k \leq d$) assume β is **s-sparse** with $s \ll d$.

- **Step 1– $S(\beta)$ support recovery**: Select the first k_1 observations (without thresholding) and compute Lasso estimator $\hat{\beta}_1$.
- **Step 2– Weight assignment**: for $j \in S(\hat{\beta}_1)$, set $\xi_j = 1$ otherwise set $\xi_j = 0$.
- **Step 3**: Apply the thresholding rule to select the remaining $k_2 = k - k_1$ observations.

Sparse Thresholding Algorithm-Lasso

Algorithm 2 Sparse Thresholding Algorithm.

Set $S_1 = \emptyset, S_2 = \emptyset$. Let $k = k_1 + k_2, n = k_1 + n_2$.
for observation $1 \leq i \leq k_1$ **do**
  Observe X^i. Choose X^i: $S_1 = S_1 \cup X^i$.
end for
Set $\gamma = 1/2, \lambda = \sqrt{4\sigma^2 \log(d)/\gamma^2 k_1}$.
Compute Lasso estimate $\hat{\beta}_1$ based on S_1, with regularization λ.
Set weights: $\xi_i = 1$ if $i \in S(\hat{\beta}_1)$, $\xi_i = 0$ otherwise.
Set $\Gamma = C\sqrt{s + 2\log(n_2/k_2)}$. Factorize $\Sigma_{S(\hat{\beta}_1)S(\hat{\beta}_1)} = UDU^T$.
for observation $k_1 + 1 \leq i \leq n$ **do**
  Observe $X^i \in \mathbf{R}^d$. Restrict to $X^i_S := X^i_{S(\hat{\beta}_1)} \in \mathbf{R}^s$.
  Compute $X^i{}_S = D^{-1/2}U^T X^i_S$.
  if $\|X^i_S\|_\xi > \Gamma$ or $k_2 - |S_2| = n - i + 1$ **then**
    Choose X^i_S: $S_2 = S_2 \cup X^i_S$.
    if $|S_2| = k_2$ **then**
      break.
    end if
  end if
end for
Return OLS estimate $\hat{\beta}_2$ based on observations in S_2.

Upper Bound- Sparse Thresholding

Theorem

Assumptions:

- $D = N(0, \Sigma)$.
- Σ , λ *and* $min_j|B_j|$ *satisfy some regularity conditions.*
- *Assume we run the Sparse Thresholding algorithm with* $k_1 = Cs\log(d)$ *observations to recover the support of* β; $C \geq 0$.

Let $\mathbf{X}_2$ *be* $k_2 = k - k_1$ *observations sampled via thresholding on* $S(\hat{\beta}_1)$. *Then for any* $\psi \in (0,1)$, *there exists* $C_1 > 0$ *and some constant* c_1 *and* c_2 *such that:*

$$Tr(\Sigma_S(\mathbf{X}^T\mathbf{X})^{-1}) \leq \frac{s}{(1-\psi)(1+\frac{2log(n_2/k_2)}{s})k_2} \tag{8}$$

with probability at least
$1 - 2exp\{-min(c_2 min(s, \log(d-s)) - \log(c_1), k_2 C_1 - log(s))\}$.

Theoretical Results

Gain:

- The performance of random sampling with lasso estimator is $O(s\log(d))$.
- For $s \ll d$, $k = \bar{C}d$ and $n = d^{\delta}$, with $\bar{C} > 0$, $\delta > 1$ the active learning bound is of order $\frac{s}{d(1+\frac{(\delta-1)\log(d)}{s})}$.
- This is a gain of at least $\log(d)$ factor with high probability over the weaker $\frac{s\log(d)}{d}$ for random sampling.

Remarks:

- The performance of the algorithm is also improved by using all k observations to fit the estimate $\hat{\beta}_2$.
- Using the thresholding algorithm to select the initial k_1 observations strongly decreases the probability of making a mistake in the support recovery.

- Recall that in order to minimize the prediction error, the best possible $\mathbf{X}^T\mathbf{X}$ is diagonal, with identical entries and trace equal to the sum of the norms.
- No selection algorithm, online or offline can do better.
- Algorithm 1 achieves this by selecting observations with large norms and uncorrelated entries(whitening).

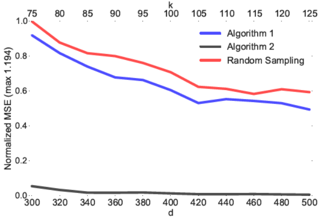
Theorem

Let A be an algorithm for the problem described. Then

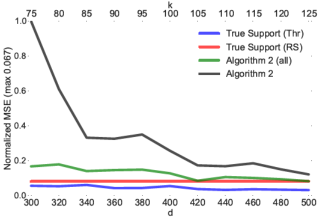
$$E_A Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \geq \frac{d^2}{E\left[\sum_{i=1}^{k} \|\bar{X}_{(i)}\|^2\right]} \geq \frac{d}{kE\left[\frac{1}{d} max_{i\in[n]} \|\bar{X}_i\|^2\right]} \tag{9}$$

Where $\bar{X}_{(i)}$ denotes the observation of the i-th largest norm. Moreover if $\mathbf{F}$ is the cdf of $max_{i\in[n]}\|\bar{X}_i\|^2$, then with probability $(1-\alpha)$,

$$Tr(\Sigma(\mathbf{X}^T\mathbf{X})^{-1}) \geq \frac{d}{k\mathbf{F}^{-1}(1-\alpha)}.$$



(a) Zooming out.



(b) Zooming in.

Figure: Sparse Linear Regression: s=7, d = 300 to 500, k=d/4 and n=7d.

- Sparse Thresholding (Algorithm 2) dramatically reduces the MSE when the true support is not know in advance.
- Algorithm 2 is still competitive even when the true support is provided.

Simulations

Non-linear Data

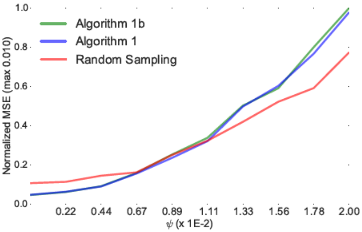


Figure: Non linear regression: model $y = \sum_i \beta_i x_i + \psi_i \sum_i x_i^2$.

- Active learning is robust to some level of non-linearity but at some point random sampling becomes more effective.

Real-World Data

Protein Structure

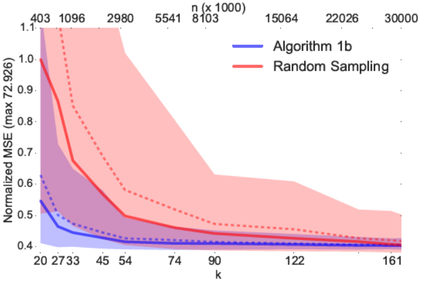


Figure: Algorithm(1b) on Protein structure data (150 iterations) : MSE of $\hat{\beta}_{OLS}$; Solid= median; Dashed= mean; Shade= Quantile confidence interval; $d = 9$; $n = 45730$

- Algorithm(1b) outperforms random sampling for all values of (n, k).

Real-World Data

Bike Sharing

Goal: To predict the number of hourly users of the service, given weather condition, including temperature, wind speed, humidity and temporal covariates.

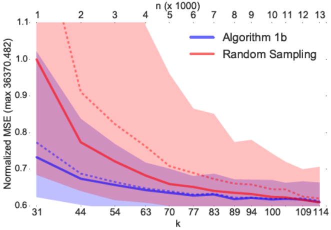


Figure: Algorithm(1b) on Bike Sharing data (300 iterations) : MSE of $\hat{\beta}_{OLS}$; $d = 12$; $n = 17379$

- The mean, median and variance of the MSE for the Algorithm(1b) estimator are smaller than that of random sampling.

Goal: To predict the year a song was released based on d= 90 covariates.

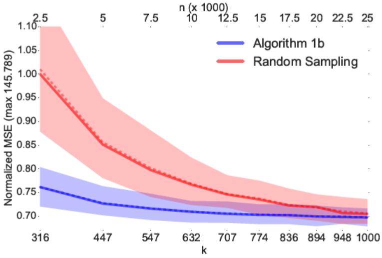


Figure: Algorithm(1b) on Song data (150 iterations) : MSE of $\hat{\beta}_{OLS}$; $d = 90$; $n = 99799$

- Algorithm(1b) improves the mean and variance of the MSE

Summary

- The algorithms proposed lead to strong improvement in MSE and variance both theoretically and empirically.
- The proposed algorithm guarantees extend to sparse linear regression in high-dimensional settings.
- The Algorithm does not perform well compare to random sampling when the budget constraint k grows large. The algorithm is therefore better suited for limited labeling budget situations.
- **Possible Extension**
    - Investigate additional robustness by combining the algorithm proposed with other approaches such as stratified sampling and random sampling in order to detect the presence of non-linearity.

Further Reading I

- Castro, Rui M., and Robert D. Nowak. "Minimax bounds for active learning." IEEE Transactions on Information Theory 54.5 (2008): 2339-2353.
- Cohn, David A., Zoubin Ghahramani, and Michael I. Jordan. "Active learning with statistical models." Journal of artificial intelligence research (1996).
- Hsu, Daniel, and Sivan Sabato. "Heavy-tailed regression with a generalized median-of-means." ICML. 2014.
- Sabato, Sivan, and Remi Munos. "Active regression by stratification." Advances in Neural Information Processing Systems. 2014.
- Settles, Burr. "Active learning literature survey." University of Wisconsin, Madison 52.55-66 (2010): 11.
- Willett, Rebecca, Robert Nowak, and Rui M. Castro. "Faster rates in regression via active learning." Advances in Neural Information Processing Systems. 2005.