COMS 6998-4 Fall 2017 Presenter: Xingyu Niu
November 13, 2017 Scribe: Jinyi Zhang

Conditional Sparse Linear Regression

1 Introduction

1.1 Background

Given a distribution D over $R^d \times R$ and data samples $(y, z) \sim D$, linear regression models the linear relationship between the independent vector y and the scalar dependent variable z. Concretely, we can find a vector $a \in R^d$ such that $\langle a, y \rangle \approx z, \forall (y, z) \sim D$ if there exists a linear relationship between y and z. Sometimes, we want to enforce the coefficients that derived from linear regression to be sparse. A sparse model may provide a good fit, as well as the interpretability of the model and utility of features for further analysis. Let's define the sparsity parameter s as the number of non-zero entries in a, which implies that $s \ll d$.

In general, there are two approaches for sparse linear regression. The first one is called Best-Subset. It selects an s-size subset of d dimensions and finds a model that provides a good fit in this dimension subset. The model learned in this way usually cannot provide a good fit for all samples. The second one is called regularized regression. We add constraints like $||a||_0 \leq s$ or $||a||_1 \leq s$ to linear regression, which makes the models sparse. In this scribe, we focus on the the methods of applying the first approach only on some segments of samples that yield good fits.

1.2 Population Segments Identification Function

A sparse linear regression model that learned by selecting a subset of dimensions usually cannot provide a good fit for all samples. However, we can certainly identify some population segments in which the sparse model fits well. This leads to the idea of conditional linear regression. We use a function $c : R^n \to \{0, 1\}$ to describe a population segment. In Juba's paper, the function c is represented by a k-DNF since it describes disjoint segments well and is efficient to calculate. The k-DNF takes binary variables as input. So we could create some "artificial variables" in $\{0, 1\}$ dependent on y. For example, we can have $x_1 = \mathbb{1}_{\{y_1 > y_2\}}$, $x_2 = \mathbb{1}_{\{y_2 y_3 < 0\}}$, and etc. Thus, the k-DNF c describes segments using $x = (x_1, x_2, ..., x_n)$ as inputs and $c(x) = 1$ indicates that the corresponding sample (y, z) is in the population segments that can be fit by a linear model.

1.3 General Settings

Given a distribution D over $\{0, 1\}^n \times R^d \times R$ and a ground-truth k-DNF c^*, we may have the following cases.

- $c^*(x) = 1$: linear model; $c^*(x) = 0$: outliers.
- $c^*(x) = 1$: linear model; $c^*(x) = 0$: another linear model.
- $c^*(x) = 1$: linear model; $c^*(x) = 0$: other complex model.

Conditional linear regression would be very suitable for these general cases as it identifies the population segments that we can learn a linear model.

2 Conditional Sparse Linear Regression

Now let's formalize the definition of conditional sparse linear regression. There is a distribution D over $\{0,1\}^n \times R^d \times R$ and samples $(x_i, y_i, z_i) \sim D, i \in [m]$, we have a hypothesis class (A, C) that A represents linear models and C represents identification functions (k-DNF). We aim to find a k-DNF c and an s-sparse model a where $(a, c) \in (A, C)$ with the following properties.

- If $c(x_i) = 0$, we won't fit/predict this sample.
- If $c(x_i) = 1$, given $\epsilon > 0$
  - If $|\langle a, y_i \rangle - z_i| \leq \epsilon$, model a provides a good fit.
  - If $|\langle a, y_i \rangle - z_i| > \epsilon$, model a provides a bad fit.

2.1 Comparison with Agnostic Selective Classification

As we have learned from previous lectures, agnostic selective classification also has a identification function g. In some sense, conditional sparse linear regression is similar to agnostic selective classification.

	Agnostic Selective Classification	Conditional Linear Regression
"Not sure" / No fit	$\Pr_{(x,y)\sim P}[g(x) = 0]$	$\Pr_{(x,y,z)\sim D}[c(x) = 0]$
Correct / Good fit	$\Pr_{(x,y)\sim P}[h(x) = y \wedge g(x) = 1]$	$\Pr_{(x,y,z)\sim D}[\lvert\langle a, y_i\rangle - z_i\rvert \leq \epsilon \wedge c(x) = 1]$
Error / Error	$\Pr_{(x,y)\sim P}[h(x) \neq y \wedge g(x) = 1]$	$\Pr_{(x,y,z)\sim D}[\lvert\langle a, y_i\rangle - z_i\rvert > \epsilon \wedge c(x) = 1]$

Table 1: Comparison between conditional sparse linear model and agnostic selective classification

2.2 Coverage & Confidence

We define the coverage and confidence of a conditional linear regression as the following two probabilities respectively.

- Coverage:
$$\Pr_{(x,y,z)\sim D}[c(x) = 1].$$
- Confidence:
$$\Pr_{(x,y,z)\sim D}[|\langle a, y\rangle - z| \leq \epsilon | c(x) = 1].$$

2.3 Problem Formulation

Given a distribution D over $\{0,1\}^n \times R^d \times R$, there exists target c^* and a^* such that

$$\text{Confidence}(c^*, a^*) = 1 \ \& \ \text{Coverage}(c^*) > \mu.$$

in realizable setting. Give the parameters $\epsilon, \mu, \delta, \gamma$, and m samples, we aim to construct an algorithm that returns the target estimates c', a' such that

$$\text{Confidence}(c', a') \geq 1 - \gamma \ \& \ \text{Coverage}(c') > \Omega\left(\left(\frac{(1-\gamma)\mu}{nd}\right)^k\right)$$

with probability at least $1 - \delta$. Moreover, we want the algorithm runs polynomial time in terms of $n, d, \frac{1}{\mu}, \frac{1}{\epsilon}, \frac{1}{\gamma}, \frac{1}{\delta}$.

2.4 Algorithm

In this section, we introduce an algorithm called Find-and-eliminate that can solve the conditional sparse linear regression problem in polynomial time. Before diving into the details, let's first define some notations.

- Given s coordinates $d_1, d_2, ..., d_s$ from $[d]$, $\prod_{d_1,...,d_s}$ is a subset-dimension projection that maps from R^d to R^s. For example, $\dot{y} = \prod_{d_1,...,d_s} y$: $\dot{y}_i = y_{d_i}, \forall i \in [s]$ means that $\dot{y}$ is the subset-dimension projection of y.
- For a set S, $\binom{n}{k}$ denote the subsets of S of size exactly k.

With the notations defined as above, the Find-and-eliminate algorithm is the following.

Algorithm 1 Find-and-eliminate algorithm

input: Examples $(x^{(1)}, y^{(1)}, z^{(1)}), ..., (x^{(m)}, y^{(m)}, z^{(m)})$, target fit ϵ, and fraction $(1-\gamma/2)\mu$

output: A k-DNF over $x_1, ..., x_n$ and linear predictor $y_1, ..., y_d$, or INFEASIBLE if none exists.

for all the $(d_1, ..., d_s) \in \binom{[d]}{s}$, $(\sigma_1, ..., \sigma_{s+1}) \in \{\pm 1\}^{s+1}$, and $(j_1, ..., j_{s+1}) \in \binom{[m]}{s+1}$ **do**
  Initialize c to be the (trivial) k-DNF over all $\binom{2n}{k}$ terms of size k.
  Let (a, ϵ') be a solution of the following linear system:

$$\langle a, \prod_{d_1,...,d_s} y^{(jl)} \rangle - z^{(jl)} = \sigma_l \epsilon' \text{ for } l = 1, 2, ..., s+1$$

  if $\epsilon' > \epsilon$ **then**
    Continue to the next iteration.
  end if
  for j = 1, 2, ..., m **do**
    if $|\langle a, \prod_{d_1,...,d_s} y^{(j)} \rangle - z^{(j)}| > \epsilon$ **then**
      for all the $T \in c$ **do**
        if $T(x^{(j)}) = 1$ **then**
          Remove T from c.
        end if
      end for
    end if
  end for
  if $\#\{j : c(x^{(j)}) = 1\} > (1-\gamma/2)\mu m$ **then**
    return a and c.
  end if
end for
return INFEASIBLE.

We can separate the algorithm into two parts. The first part is to find an estimate of a^* and the second part is to find an estimate of c^*. The algorithm is actually a brute-force search over all possible estimates of a^*. The linear system in the algorithm is derived from an optimization problem. After solving the linear system, the algorithm ensures that the regression error less than or equal to ϵ. Concretely, we have the real risk of conditional linear regression as $\Pr_{(x,y,z)\sim D}[|\langle a, y\rangle - z| > \epsilon \wedge c(x) = 1]$. Given the samples M, we have the empirical risk as $\Pr_M[|\langle a, y\rangle - z| > \epsilon \wedge c(x) = 1]$. If the algorithm returns (a', c'), then for $(x_i, y_i, z_i) \in M$ we have $|\langle a, y_i\rangle - z_i| > \epsilon \wedge c'(x_i) = 0$. Thus, the empirical risk of (a', c') is 0. The algorithm is essentially to return the empirical risk minimizer of conditional linear regression.

Theorem 1. *Suppose that D is a joint probability distribution over $x \in 0, 1^n$, $y \in R^d$, and*

$z \in R$ such that there is a k-DNF c for which for some s-sparse $a \in R^d$

$$\Pr_{(x,y,z)\sim D}[|\langle a, y_i\rangle - z_i| \leq \epsilon | c(x) = 1] = 1 \text{ and } \Pr_{(x,y,z)\sim D}[c(x) = 1] \geq \mu$$

Then given ϵ, μ, and δ in $(0, 1)$ and $\gamma \in (0, 1/2]$ and

$$m = O\left(\frac{1}{\mu\gamma}\left(s \log s + s \log d + n^k + \log \frac{1}{\delta}\right)\right)$$

examples from D, for any constants s and k, Algorithm 1 runs in polynomial time in n, d, and m and finds an s-sparse a' and k-DNF c' such that

$$\Pr_{(x,y,z)\sim D}[|\langle a', y_i\rangle - z_i| \leq \epsilon | c'(x) = 1] \geq 1 - \gamma \text{ and } \Pr_{(x,y,z)\sim D}[c'(x) = 1] \geq (1 - \gamma)\mu$$

with probability $1 - \delta$.

2.4.1 Preliminaries

Before we prove Theorem 1, let's recall two key results from previous lectures.

1. Multiplicative Chernoff Tail Bound. Let random variable $X = \sum_{i=1}^{n} X_i$ and all binary variables are independent. Let $P_i = \Pr[X_i = 1]$ and $\bar{M} = E[X] = \sum_{i=1}^{M} P_i$, we have

   $$\Pr[X < (1 - \delta)\bar{M}] \leq e^{-\bar{M}\delta^2/2}$$

   for $0 < \delta < 1$.

2. Sample complexity theorem. Given hypothesis class H and its real risk minimizer of h^* with $\text{err}(h^*) = 0$. Let $\hat{h}$ be the empirical risk minimizer of H given m examples and its empirical risk $\text{err}_m(\hat{h}) = 0$. For $\forall \epsilon, \delta > 0$, if the sample complexity

   $$m = O\left(\frac{VC(H)\log\frac{1}{\epsilon} + \log\frac{1}{\delta}}{\delta}\right),$$

   then we have $\text{err}(\hat{h}) - \text{err}(h^*) \leq \epsilon$ with probability at least $1 - \delta$.

2.4.2 Proof of Theorem 1

Proof. First we show the time complexity of the algorithm. The outer loop runs in $O(d^s m^{s+1})$ obviously. And the inner loop runs in $O(mn^k)$ since we have m samples and $\binom{2n}{k}$ different k-DNF terms. In total, the algorithm runs in $O(d^s m^{s+2} n^k)$.

Then we prove Theorem 1 by answering the following two questions.

1. Is the target hypothesis (a^*, c^*) in the solution set if the algorithm returns all feasible solutions? (completeness)

2. Do all the solutions returned by the algorithm satisfy the confidence and coverage requirement? (correctness)

Before look into these two questions, we first explain why the algorithm solves the linear system. For the given m samples, we define $\bar{S}$ such that $(x_i, y_i, z_i) \in \bar{S}$ and we have $c^*(x_i) = 1$ and $|\langle a^*, y_i\rangle - z_i| \leq \epsilon$. Of course $\bar{S}$ is unknown. Consider the following optimizing problem.

$$\min_{a,\epsilon} \epsilon$$

$$\text{s.t. } -\epsilon \leq \langle a, y_i\rangle - z_i \leq \epsilon, (x_i, y_i, z_i) \in \bar{S}$$

It can be reformulated as a linear system as the following.

$$\min_{a,\epsilon} (a, \epsilon)^T (\mathbf{0}, 1)$$

$$\text{s.t. } (a, \epsilon)^T (y_i, -1) \leq z_i, (a, \epsilon)^T (-y_i, -1) \leq -z_i, \epsilon \geq 0, \text{and } (x_i, y_i, z_i) \in \bar{S}$$

This system has $s + 1$ variables and can be solved by linear programming. Its objective function is linear and convex. The optimal solution lies on one of the vertex of the polytone defined by the constraints. Thus, we can brute force $\binom{\bar{S}}{s+1}$ equations and solve them, and a^* would be one of the solutions. However, we do not know $\bar{S}$. Thus we have to brute force all $\binom{m}{s+1}$ possible supports since $\binom{\bar{S}}{s+1} \subseteq \binom{m}{s+1}$. This indicates that a^* is a feasible solution to the linear system and it's possible to get empirical risk zero by the algorithm. Now we are close to answer the first question. Assume the algorithm returns a^* exactly and its corresponding identification function h^*, which could be a much broader version of c^*. What the algorithm essentially does is that for all samples $(x_j, y_j, z_j) \in M$, if $|\langle a^*, y_j\rangle - z_j| > \epsilon$, it eliminates the term $T \in h^*$ such that $T(x_j) = 1$, which implies that $(x_j, y_j, z_j) \notin \bar{S}$. Then for any term $T' \in c^*$, we will not eliminate it in h^*. So we have $\text{Terms}(c^*) \subseteq \text{Terms}(h^*)$. For all samples (x_i, y_i, z_i) in $\bar{S}$, we have $c^*(x_i) = h^*(x_i) = 1$. If the number of $h^*(x_i) = 1$ greater than or equal to $(1 - \gamma/2)\mu m$, then h^* would be returned by the algorithm, which implies that c^* is a feasible solution of the algorithm.

Let's consider the probability of $|S| \geq (1 - \gamma/2)\mu m$. First, we define

$$Y_i = \mathbb{1}_{\{c^*(x)=1\}}, \text{and } Y = \sum_{i=1}^{m} Y_i.$$

Let $p := \Pr[Y_i = 1] \geq \mu$, we have $E[Y] = pm \geq \mu m$. We set $\delta = \frac{\gamma}{2}$ and apply Chernoff bound, so we have

$$\Pr\left[Y < \left(1 - \frac{\gamma}{2}\right) pm\right] \leq \exp\left(-\frac{pm\gamma^2}{\delta}\right) \leq \exp\left(-\frac{\mu m\gamma^2}{\delta}\right).$$

If we substitute $m \geq \frac{4}{\mu\gamma^2} \log \frac{3}{\delta}$ into the inequality, then

$$\Pr\left[Y < \left(1 - \frac{\gamma}{2}\right) pm\right] \leq \frac{\delta}{3}.$$

Therefore, we have

$$\Pr\left[Y \geq \left(1 - \frac{\gamma}{2}\right)\mu m\right] \geq 1 - \frac{\delta}{3}.$$

It implies that the algorithm returns a solution with probability at least $1 - \frac{\delta}{3}$. This answers the first question mentioned as above.

Now we show the correctness of the algorithm that is for every solution returned by the algorithm, the solution satisfies confidence and coverage requirements. Suppose the empirical risk minimizer (a', c') with $\mathrm{err}_M(a', c') = 0$ is returned by the algorithm and the target (a^*, c^*) has zero real risk (realizable setting), we want to bound the real risk of (a', c') under ϵ. Concretely,

$$\mathrm{err}(a', c') - \mathrm{err}(a^*, c^*) \leq \epsilon.$$

Since $\mathrm{err}(a^*, c^*) = 0$, we have

$$\mathrm{err}(a', c') \leq \epsilon.$$

By applying sample complexity theorem, we set

$$\epsilon = \frac{\mu\gamma}{2} \text{ and } m = O\left(\frac{VC(A, C) + \log\frac{1}{\delta}}{\mu\gamma}\right)$$

to achieve the above error bound. Now we need to determine the VC-dimension of (A, C). Since C is the k-DNFs over all $\binom{2n}{k}$ terms of size k, the number of possible k-DNFs is $2^{\binom{2n}{k}}$. Therefore,

$$VC(C) = O\left(\binom{2n}{k}\right) = O(n^k).$$

Moreover, the VC-dimension of a linear threshold function in dimension s is $s + 1$. We have $\binom{s}{d} \leq d^s$ supports in total. The union of them is A. Thus,

$$VC(A) = O(\log d^s + (s + 1)\log(s + 1)).$$

Therefore,

$$VC(A, C) = O(s \log s + s \log d + n^k).$$

Since $\epsilon = \frac{\mu\gamma}{2}$, with m samples and probability $1 - \delta$, we have the real risk

$$\mathrm{err}(a', c') = \Pr_{(x,y,z)\sim D}[\langle a', y\rangle - z > \epsilon \wedge c'(x) = 1] \leq \frac{\mu\gamma}{2}.$$

Since the algorithm returns a solution if the number of samples in the population segment is larger than or equal to $(1 - \gamma/2)\mu m$, we have the probability that a sample is in the population segment as

$$\Pr_M[c'(x) = 1] \geq \left(1 - \frac{\gamma}{2}\right)\mu.$$

By applying Bernstein's inequality, we have the coverage as

$$\Pr[c'(x) = 1] \geq \frac{1 - \frac{\gamma}{2}}{1 + \frac{\gamma}{2}}\mu \geq (1 - \gamma)\mu.$$

Then, the confidence is

$$\Pr_{(x,y,z)\sim D}[|\langle a, y\rangle - z| \leq \epsilon | c(x) = 1] = \frac{\Pr[c'(x) = 1]}{\frac{\mu m}{2}} \geq 1 - \gamma.$$

Therefore, both coverage and confidence satisfy the requirement. □

3 Conditional Distribution Search

In condition linear regression, we aim to learn a^* and c^* jointly. Sometimes, we may be only interested in c^*. Learning the estimate of c^* is called conditional distribution search.

Concretely, we formally define the problem as the following. We denote the binary label as b. Given a distribution D over $\{0,1\}^n \times \{0,1\}$, there exists a target c^* such that

$$\Pr_{(x,b)\in D}[b = 1 | c^*(x) = 1] = 1 \text{ and } \Pr_{(x,b)\in D}[c^*(x) = 1] > \mu.$$

in realizable setting. Give the parameters μ, δ, γ, and m samples, we aim to construct an algorithm that returns the target estimate c' such that

$$\Pr_{(x,b)\in D}[b = 1 | c^*(x) = 1] = 1 - \gamma \text{ and } \Pr_{(x,b)\in D}[c^*(x) = 1] > \Omega\left(\left(\frac{(1-\gamma)\mu}{n}\right)^k\right)$$

with probability at least $1 - \delta$. Moreover, we want the algorithm runs polynomial time in terms of $n, \frac{1}{\mu}, \frac{1}{\gamma}, \frac{1}{\delta}$.

Theorem 2. *For any algorithm that provides feasible solutions for conditional linear regression, it also provides feasible solutions to conditional distribution search.*

Bibliographic notes

Conditional sparse linear regression and all its analysis is due to Juba [1]. Conditional distribution search and all its analysis is also due to Juba [1].

References

[1] Brendan Juba. Conditional Sparse Linear Regression. *at arXiv*, 2016.