How to Estimate Mixture Logit Propensity Score

Applying Adaboost

Junchen Feng

Harris School of Public Policy Studies
University of Chicago

January 14, 2013

Motivation

- Propensity score matching is magic,
  but how do you get the propensity score?

Motivation

- Propensity score matching is magic,
  but how do you get the propensity score?
- Hirano, Imbens, Ridder(2003) promised a solution in the asymptopia
- But asymptopia is difficult to reach and in real world, polynomial logit does not work well.

Motivation

- Propensity score matching is magic,
  but how do you get the propensity score?
- Hirano, Imbens, Ridder(2003) promised a solution in the asymptopia

- Adaboost algorithm has superb performance under finite sample for the tails

Motivation

- Propensity score matching is magic,
  but how do you get the propensity score?
- Hirano, Imbens, Ridder(2003) promised a solution in the asymptopia

- Adaboost algorithm has superb performance under finite sample for the tails
- Yet much work needs to be done before one can confidently use Adaboost

Motivation

- Propensity score matching is magic,
  but how do you get the propensity score?
- Hirano, Imbens, Ridder(2003) promised a solution in the asymptopia

- Adaboost algorithm has superb performance under finite sample for the tails
- Yet much work needs to be done before one can confidently use Adaboost
- We can learn a lot from the latest fad in the machine learning community.

Paradise in the Asymptopia

- I do not claim that I fully understand the math in the appendix of Hirano, Imbens, Ridder(2003)

Paradise in the Asymptopia

- I do not claim that I fully understand the math in the appendix of Hirano, Imbens, Ridder(2003)
- According to Hirano's class slide, the intuition is this

Paradise in the Asymptopia

- I do not claim that I fully understand the math in the appendix of Hirano, Imbens, Ridder(2003)
- According to Hirano's class slide, the intuition is this
  - Let $p(Y=1|X) = m(x) = \frac{1}{1+e^{-k(x)}} = \Lambda(k(x))$.

Paradise in the Asymptopia

- I do not claim that I fully understand the math in the appendix of Hirano, Imbens, Ridder(2003)
- According to Hirano's class slide, the intuition is this
  - Let $p(Y = 1|X) = m(x) = \frac{1}{1+e^{-k(x)}} = \Lambda(k(x))$.
  - We can approximate m(x) to any degree by a polynomials, but the sum may not be bounded between $[0, 1]$.

Paradise in the Asymptopia

- I do not claim that I fully understand the math in the appendix of Hirano, Imbens, Ridder(2003)
- According to Hirano's class slide, the intuition is this
  - Let $p(Y = 1|X) = m(x) = \frac{1}{1+e^{-k(x)}} = \Lambda(k(x))$.
  - We can approximate m(x) to any degree by a polynomials, but the sum may not be bounded between $[0, 1]$.
  - Since $\Lambda^{-1}(x) = -log(\frac{1}{y} - 1)$ is invertible, there is a definite mapping between k(x) and m(x).

Paradise in the Asymptopia

- I do not claim that I fully understand the math in the appendix of Hirano, Imbens, Ridder(2003)
- According to Hirano's class slide, the intuition is this
  - Let $p(Y = 1|X) = m(x) = \frac{1}{1+e^{-k(x)}} = \Lambda(k(x))$.
  - We can approximate m(x) to any degree by a polynomials, but the sum may not be bounded between $[0, 1]$.
  - Since $\Lambda^{-1}(x) = -log(\frac{1}{y} - 1)$ is invertible, there is a definite mapping between k(x) and m(x).
  - Instead of approximating m(x), we can approximate k(x) arbitrarily well by polynomials and then convert to m(x).

Paradise Lost

- Problem No.1:
  Mixture Logit usually does not look like logit function!
  Thus require high order polynomials.
  Consider

$$P(Y=1) = 0.5 \times \frac{1}{1+e^{-1-0.5x}} + 0.5 \times \frac{1}{1+e^{-1-10x}}$$

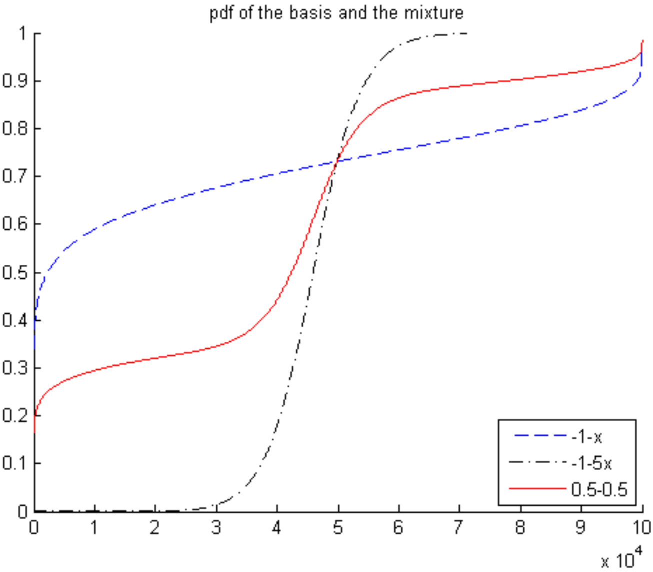

pdf of the basis and the mixture
1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
1
2
3
4
5
6
7
8
9
10
x 10^4
-1-x
-1-5x
0.5-0.5

Figure : CDF

Paradise Lost

- Problem No.1:
  Mixture Logit usually does not look like logit function!
  Thus require high order polynomials.
  Consider

$$P(Y=1) = 0.5 \times \frac{1}{1+e^{-1-0.5x}} + 0.5 \times \frac{1}{1+e^{-1-10x}}$$

Paradise Lost

- Problem No.1:
  Mixture Logit usually does not look like logit function!
  Thus require high order polynomials.
  Consider

$$P(Y=1) = 0.5 \times \frac{1}{1+e^{-1-0.5x}} + 0.5 \times \frac{1}{1+e^{-1-10x}}$$

- Problem No.2:
  When order of polynomials increases, the numerical stability becomes questionable, if you ever reaches 9th polynomials.
  This problem can possibly be solved by Kernel Support Vector Machine, but I will leave to another day.

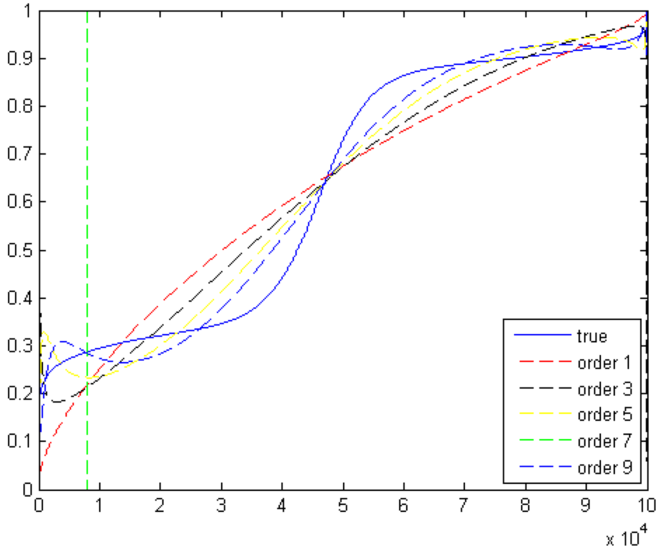


Figure : Polynomial Logit fit

What about Histogram

- Adaboost will sound like histogram, but they are not similar.

What about Histogram

- Adaboost will sound like histogram, but they are not similar.
- To see this, I will present the histogram estimator first.

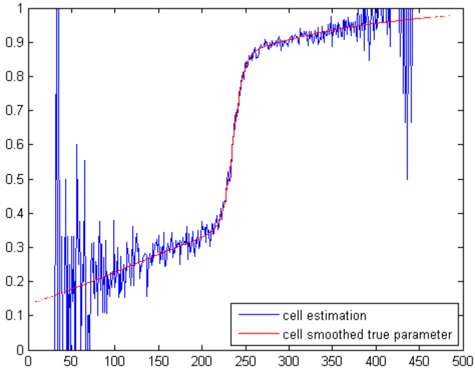


Figure : Histogram fit

What about Histogram

- Adaboost will sound like histogram, but they are not similar.
- To see this, I will present the histogram estimator first.

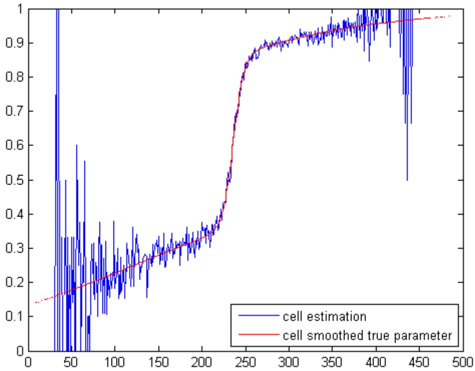


Figure : Histogram fit

- As you can see, terrible performance at the tail.

What is Adaboost?

- Boost the learning capability of the base learners, which can be quite stupid (and usually is...)

What is Adaboost?

- Boost the learning capability of the base learners, which can be quite stupid (and usually is...)
- Re-sample the data to generate weights for the base learner.

What is Adaboost?

- Boost the learning capability of the base learners,
  which can be quite stupid (and usually is...)
- Re-sample the data to generate weights for the base learner.
- Procedure of Adaboost
  - (1) Under weight w_{t-1}, find the optimal base learner $f_t(x)$

What is Adaboost?

- Boost the learning capability of the base learners, which can be quite stupid (and usually is...)
- Re-sample the data to generate weights for the base learner.
- Procedure of Adaboost
  - (1) Under weight w_{t-1}, find the optimal base learner $f_t(x)$
  - (2) Calculate the average error $e_t = E(1[y_i \neq f_t(x_i)])$.

What is Adaboost?

- Boost the learning capability of the base learners, which can be quite stupid (and usually is...)
- Re-sample the data to generate weights for the base learner.
- Procedure of Adaboost
  - (1) Under weight w_{t-1}, find the optimal base learner $f_t(x)$
  - (2) Calculate the average error $e_t = E(1[y_i \neq f_t(x_i)])$.
  - (3) Calculate the classifier weight $a_t = log(\frac{(1-e_t)}{e_t})$

What is Adaboost?

- Boost the learning capability of the base learners, which can be quite stupid (and usually is...)
- Re-sample the data to generate weights for the base learner.
- Procedure of Adaboost
  - (1) Under weight w_{t-1}, find the optimal base learner $f_t(x)$
  - (2) Calculate the average error $e_t = E(1[y_i \neq f_t(x_i)])$.
  - (3) Calculate the classifier weight $a_t = log(\frac{(1-e_t)}{e_t})$
  - (4) Update the date weight $w_{(t,i)} = w_{(t-1,i)} e^{a_t 1[y_i \neq f_t(x_i)]}$

    Normalize $\sum_{i=1}^{N} w_{t,i} = 1$

What is Adaboost?

- Boost the learning capability of the base learners, which can be quite stupid (and usually is...)
- Re-sample the data to generate weights for the base learner.
- Procedure of Adaboost
  - (1) Under weight w_{t-1}, find the optimal base learner $f_t(x)$
  - (2) Calculate the average error $e_t = E(1[y_i \neq f_t(x_i)])$.
  - (3) Calculate the classifier weight $a_t = log(\frac{(1-e_t)}{e_t})$
  - (4) Update the date weight $w_{(t,i)} = w_{(t-1,i)} e^{a_t 1[y_i \neq f_t(x_i)]}$

    Normalize $\sum_{i=1}^{N} w_{t,i} = 1$
  - repeat (1)-(4) until satisfied (or bored).

    output classification $\hat{y}_i = sign(F(x_i)) = sign(\sum_{t=1}^{T} a_t f_t(x_i))$

It's Density, Stupid

- Friedman, Hastie & Tibshirani(2000) points out that Adaboost can be thought as an approximated logit regression

$$P(Y = 1|X) = \frac{1}{1 + e^{-2F(x)}} = \frac{1}{1 + e^{-2(\sum_{t=1}^{T} a_t f_t(X))}}$$

- It works very well on this example

Performance of Adaboost

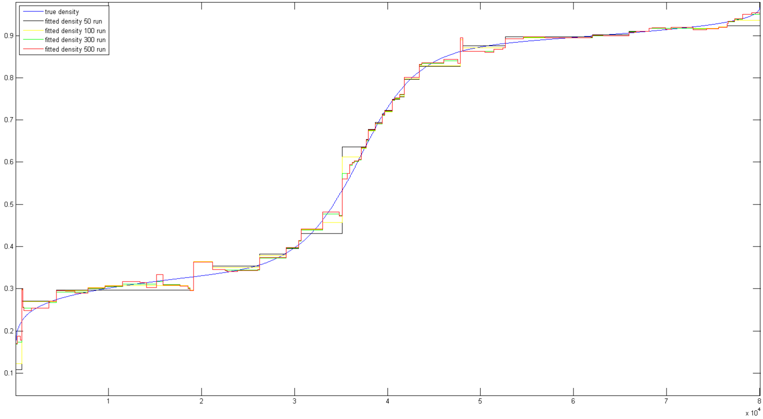


Figure : Adaboost fit

Behavior of Adaboost

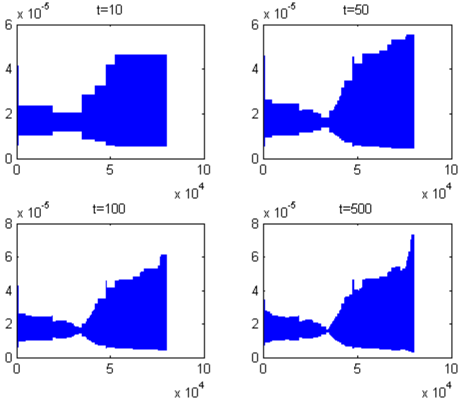


Figure : Weights on the base learners

Further Reserach Topic

- Can we analytically bound the risk of the Adaboost estimator of a mixture logit density?
  Or in plain English: Why it works?

Further Reserach Topic

- Can we analytically bound the risk of the Adaboost estimator of a mixture logit density?
  Or in plain English: Why it works?
- Adaboost does not over fit if the data is **RIGHT**.
  What happens if the data has measurement errors?

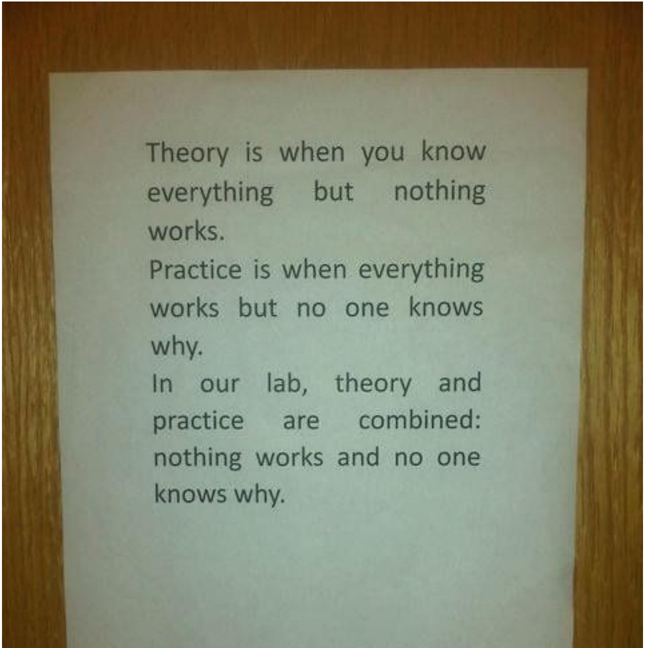
Further Reserach Topic

- Can we analytically bound the risk of the Adaboost estimator of a mixture logit density?
  Or in plain English: Why it works?
- Adaboost does not over fit if the data is **RIGHT**.
  What happens if the data has measurement errors?
- How does it do in higher dimension?
  It is likely to suffer from curse of dimension, but how severe?

Further Reserach Topic

- Can we analytically bound the risk of the Adaboost estimator of a mixture logit density?
  Or in plain English: Why it works?
- Adaboost does not over fit if the data is **RIGHT**.
  What happens if the data has measurement errors?
- How does it do in higher dimension?
  It is likely to suffer from curse of dimension, but how severe?
- How does it do if we have missing X?
  To be fair, for propensity matching, CIA needs to hold.

Q&A


Theory is when you know everything but nothing works.
Practice is when everything works but no one knows why.
In our lab, theory and practice are combined: nothing works and no one knows why.