

# A Nonparametric Maximum Likelihood Approach to Mixture of Regressions



Hansheng Jiang



Aditya Guntuboyina

University of California, Berkeley

IISA Student Paper Competition

July 18, 2020

# Contents

- 1 Introduction
- 2 Fitting MLR with NPMLE
- 3 Existence and Computation
- 4 Finite-sample Hellinger Error Bound
- 5 Summary

# Contents

## 1 Introduction

## 2 Fitting MLR with NPMLE

## 3 Existence and Computation

## 4 Finite-sample Hellinger Error Bound

## 5 Summary

# Background

- Mixture models are useful for analysis in heterogeneous populations
- Mixture of linear regressions (MLR) is a popular mixture model and has a long history (Quandt, 1958)
- MLR is also known as the Hierarchical Mixture of Experts model (Jordan and Jacobs, 1994) in the machine learning community

# Background

- Mixture models are useful for analysis in heterogeneous populations
- Mixture of linear regressions (MLR) is a popular mixture model and has a long history (Quandt, 1958)
- MLR is also known as the Hierarchical Mixture of Experts model (Jordan and Jacobs, 1994) in the machine learning community

## Applications

- Medicine and pharmacokinetics (Lai and Shih, 2003)
- Health care (Deb and Holmes, 2000)
- Marketing and business (Wedel and Kamakura, 2012)

# The Mixture of Linear Regressions (MLR) Model

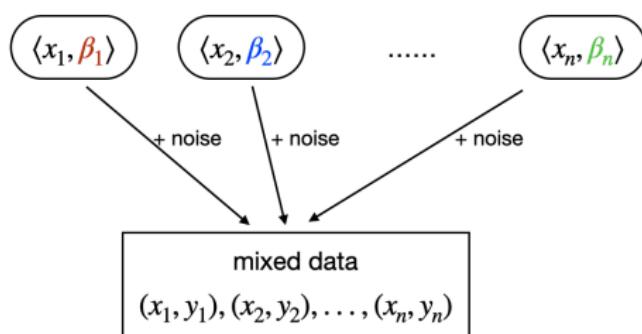
MLR model with unknown mixing probability measure  $G^*$

$$Y_i = x_i^T \beta_i + Z_i \quad \text{with} \quad Z_1, \dots, Z_n \stackrel{i.i.d.}{\sim} N(0, \sigma^2)$$

where  $\sigma > 0$  and

$$\beta_1, \dots, \beta_n \stackrel{i.i.d.}{\sim} G^*$$

for an unknown probability measure  $G^*$  on  $\mathbb{R}^p$ , and  $G^*$  is independent of  $Z_1, \dots, Z_n$

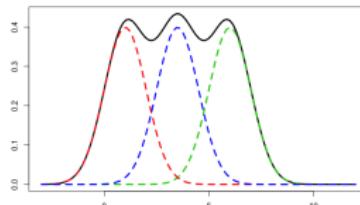


## Problem statement

Given data  $(x_1, y_1), \dots, (x_n, y_n)$  with  $x_i \in \mathbb{R}^p$  and  $y_i \in \mathbb{R}$ , we want to nonparametrically estimate  $G^*$

# Related Work

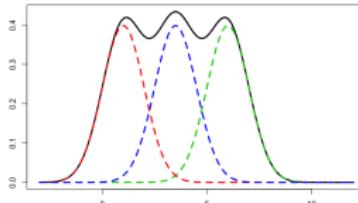
- Gaussian location mixture



- Finite mixture of linear regression models with  $k$  components
  - ▶ The finite formulation is non-convex
  - ▶ Commonly estimated via Expectation-Maximization algorithm

# Related Work

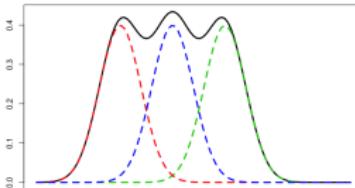
- Gaussian location mixture



- Finite mixture of linear regression models with  $k$  components
  - ▶ The finite formulation is non-convex
  - ▶ Commonly estimated via Expectation-Maximization algorithm
- Machine learning papers on finite-component mixture of linear regression (Li and Liang, 2018), high-dimensional Gaussian mixture (Yi and Caramanis, 2015)

# Related Work

- **Gaussian location mixture**



- **Finite mixture of linear regression models with  $k$  components**

- ▶ The finite formulation is non-convex
- ▶ Commonly estimated via Expectation-Maximization algorithm

- Machine learning papers on finite-component mixture of linear regression (Li and Liang, 2018), high-dimensional Gaussian mixture (Yi and Caramanis, 2015)

Previous nonparametric approaches to MLR

- Beran and Hall (1992), Beran and Millar (1994), Beran et al. (1996)
- Hoderlein et al. (2010)

We propose nonparametric maximum likelihood approach to the MLR model

# Contents

1 Introduction

2 Fitting MLR with NPMLE

3 Existence and Computation

4 Finite-sample Hellinger Error Bound

5 Summary

# NPMLE of MLR

Under the MLR model, the conditional density  $f_{x_i}^G$  of  $Y_i$  given  $x_i$  is

$$f_{x_i}^G(y_i) = \frac{1}{\sigma} \int \phi\left(\frac{y_i - x_i^\top \beta}{\sigma}\right) dG(\beta), i = 1, \dots, n$$

# NPMLE of MLR

Under the MLR model, the conditional density  $f_{x_i}^G$  of  $Y_i$  given  $x_i$  is

$$f_{x_i}^G(y_i) = \frac{1}{\sigma} \int \phi\left(\frac{y_i - x_i^\top \beta}{\sigma}\right) dG(\beta), i = 1, \dots, n$$

## Definition

The nonparametric maximum likelihood estimator (NPMLE)  $\hat{G}$  of the true mixing probability measure  $G^*$  in the MLR model is defined by

$$\hat{G} \in \arg \max_G \sum_{i=1}^n \log f_{x_i}^G(y_i),$$

where the arg max is over all probability measures supported on some set  $K$  in  $\mathbb{R}^p$

# NPMLE of MLR

Under the MLR model, the conditional density  $f_{x_i}^G$  of  $Y_i$  given  $x_i$  is

$$f_{x_i}^G(y_i) = \frac{1}{\sigma} \int \phi\left(\frac{y_i - x_i^\top \beta}{\sigma}\right) dG(\beta), i = 1, \dots, n$$

## Definition

The nonparametric maximum likelihood estimator (NPMLE)  $\hat{G}$  of the true mixing probability measure  $G^*$  in the MLR model is defined by

$$\hat{G} \in \arg \max_G \sum_{i=1}^n \log f_{x_i}^G(y_i),$$

where the arg max is over all probability measures supported on some set  $K$  in  $\mathbb{R}^p$

- This is a convex optimization in terms of the **likelihood vector**  
 $f = (f_{x_1}^G(y_1), \dots, f_{x_n}^G(y_n))$

# Contents

- 1 Introduction
- 2 Fitting MLR with NPMLE
- 3 Existence and Computation
- 4 Finite-sample Hellinger Error Bound
- 5 Summary

# Existence of NPMLE

## Theorem

*For MLR model, if the maximization search space  $K$  in NPMLE is the whole space  $\mathbb{R}^p$ , or a compact set in  $\mathbb{R}^p$ , then there exists an NPMLE that is supported on at most  $n$  points in set  $K$ .*

# Existence of NPMLE

## Theorem

*For MLR model, if the maximization search space  $K$  in NPMLE is the whole space  $\mathbb{R}^p$ , or a compact set in  $\mathbb{R}^p$ , then there exists an NPMLE that is supported on at most  $n$  points in set  $K$ .*

- Previous results are only shown for compact sets (Lindsay, 1983)

# Existence of NPMLE

## Theorem

For MLR model, if the maximization search space  $K$  in NPMLE is the whole space  $\mathbb{R}^p$ , or a compact set in  $\mathbb{R}^p$ , then there exists an NPMLE that is supported on at most  $n$  points in set  $K$ .

- Previous results are only shown for compact sets (Lindsay, 1983)

## Corollary

For any NPMLE  $\hat{G}$ ,  $f^{\hat{G}} = (f_{x_1}^{\hat{G}}, \dots, f_{x_n}^{\hat{G}})^T$  is the unique optimal solution to

$$\begin{aligned} & \text{maximize} \quad L(f) = \frac{1}{n} \sum_{i=1}^n \log f(i) \\ & \text{subject to} \quad f \in \text{conv}(\mathcal{P}_K) \end{aligned}$$

Here  $\mathcal{P}_K = \{f^\beta : \beta \in K\}$ ,  $\text{conv}(\cdot)$  represents convex hull

# Brief Intro to Conditional Gradient Method (CGM)

- Conditional gradient method (also known as Frank-Wolfe algorithm) (Frank and Wolfe, 1956)
- It is an iterative algorithm for constrained convex optimization
- Recently regained attention due to its efficiency in large scale data analysis (Jaggi, 2013)

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$

**Input:** Noise level  $\sigma$ , search space  $K$

Initialization: likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$

**while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$
3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$
4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$ **Input:** Noise level  $\sigma$ , search space  $K$ **Initialization:** likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$ **while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$
3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$
4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$

**Input:** Noise level  $\sigma$ , search space  $K$

Initialization: likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$

**while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$
3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$
4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$

**Input:** Noise level  $\sigma$ , search space  $K$

Initialization: likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$

**while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$
3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$
4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$

**Input:** Noise level  $\sigma$ , search space  $K$

Initialization: likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$

**while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$

3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$

4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$

**Input:** Noise level  $\sigma$ , search space  $K$

Initialization: likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$

**while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$
3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$
4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Computing NPMLE by CGM

---

**Algorithm 1:** Conditional gradient method for NPMLE

---

**Data:**  $\{(x_i, y_i)\}_{i=1}^n$

**Input:** Noise level  $\sigma$ , search space  $K$

Initialization: likelihood vector  $f^{(0)} = f^{\beta_0}$ , active set  $\mathcal{A}^{(0)} = \{f^{\beta_0}\}$

**while** stopping criterion not met **do**

1. Approximately solving subproblem: Find  $\tilde{g}^{(t)} \in \mathcal{P}_K$  s.t.

$$\langle \tilde{g}^{(t)}, \nabla L(f^{(t)}) \rangle \geq \max_{g \in \mathcal{P}_K} \langle g, \nabla L(f^{(t)}) \rangle - \epsilon_s = \max_{g \in \mathcal{A}} \sum_{i=1}^n \frac{g(i)}{f^{(t)}(i)} - \epsilon_s$$

2. Adding the new vector to active set:  $\mathcal{A}^{(t+1)} = \mathcal{A}^{(t)} \cup \{\tilde{g}^{(t)}\}$
3. Re-optimization:  $f^{(t+1)} := \arg \max_{f \in \text{conv}(\mathcal{A}^{(t+1)})} L(f)$
4. Updating active set:  $\mathcal{A}^{(t+1)} = \{g_j^{(t+1)} | \pi_j^{(t+1)} > 0\}$  for  
 $f^{(t+1)} = \sum_{i=1}^{N_{t+1}} \pi_j^{(t+1)} g_j^{(t+1)}$

**end**

---

# Why Conditional Gradient Method ?

- **Discretization-free**

Instead of discretization, CGM adaptively adds new points into the support of the estimator

# Why Conditional Gradient Method ?

- **Discretization-free**

Instead of discretization, CGM adaptively adds new points into the support of the estimator

- **Convergence guarantee**

CGM for NPMLE has  $O(\frac{1}{T})$  convergence rate under certain assumptions

# Why Conditional Gradient Method ?

- **Discretization-free**

Instead of discretization, CGM adaptively adds new points into the support of the estimator

- **Convergence guarantee**

CGM for NPMLE has  $O(\frac{1}{T})$  convergence rate under certain assumptions

- **Efficiency and practicality**

The only computational bottleneck in CGM is the solving subproblem step

- ▶ It suffices to do this step approximately, and the re-optimization step makes sure the likelihood function does not decrease
- ▶ We use off-the-shelf solver for this step and achieves satisfactory numerical performances (see numerical examples later)

# Why Conditional Gradient Method ?

- **Discretization-free**

Instead of discretization, CGM adaptively adds new points into the support of the estimator

- **Convergence guarantee**

CGM for NPMLE has  $O(\frac{1}{T})$  convergence rate under certain assumptions

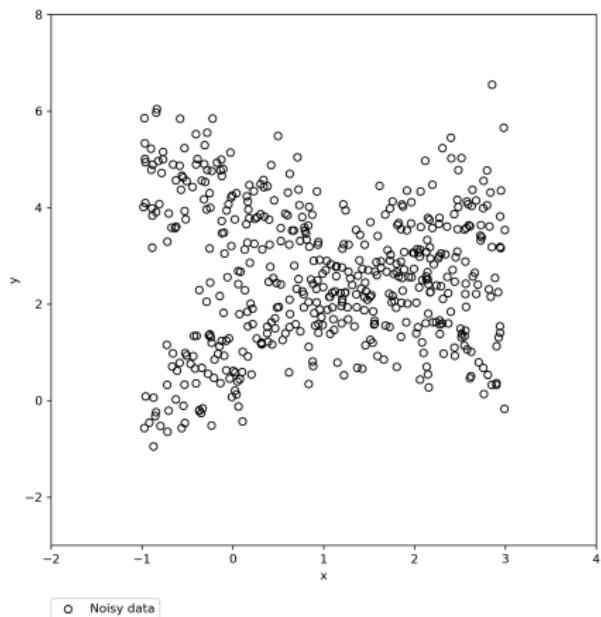
- **Efficiency and practicality**

The only computational bottleneck in CGM is the solving subproblem step

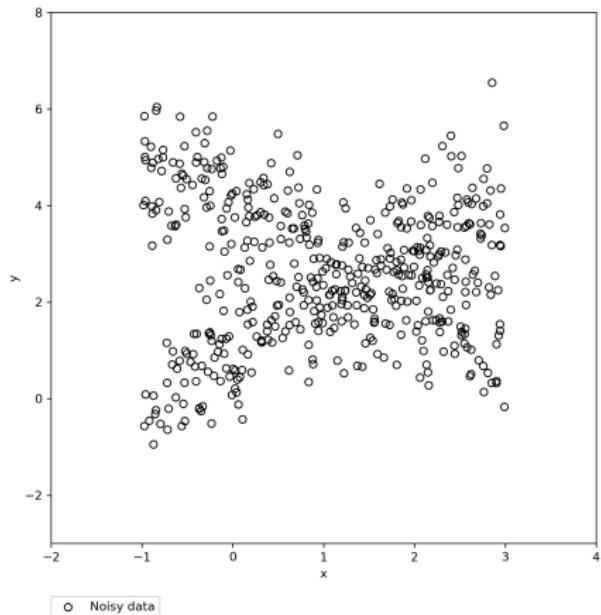
- ▶ It suffices to do this step approximately, and the re-optimization step makes sure the likelihood function does not decrease
- ▶ We use off-the-shelf solver for this step and achieves satisfactory numerical performances (see numerical examples later)

- Related to vertex direction method from the optimal design literature (Wu, 1978)

# How many components are there?



# How many components are there?



- NPMLE is agnostic to the “number” of components

# Numerical Example 1

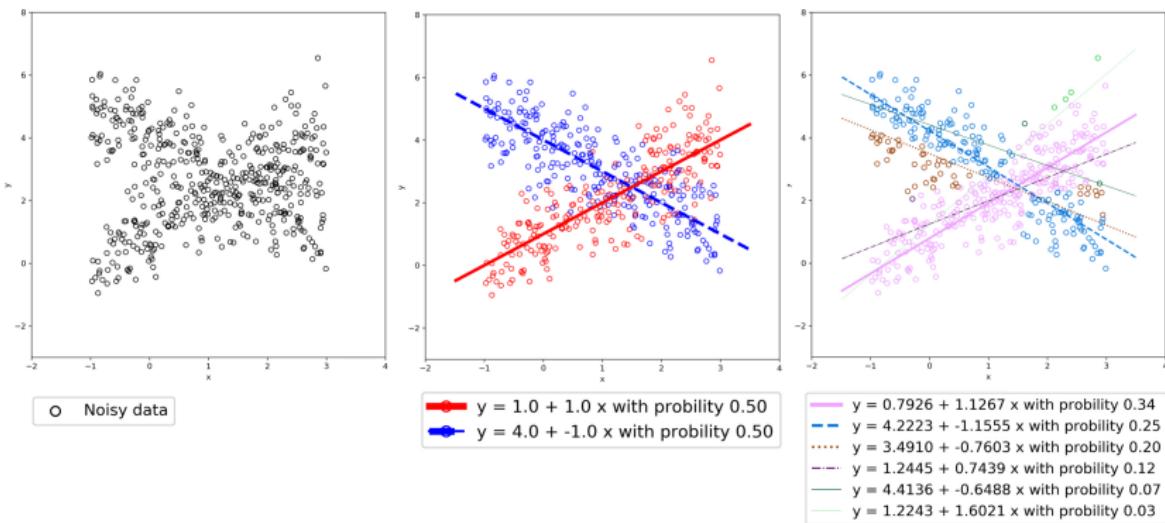


Figure: Left: Noisy data; Middle: True mixture; Right: Fitted mixture

# Numerical Example 1 (Continued)

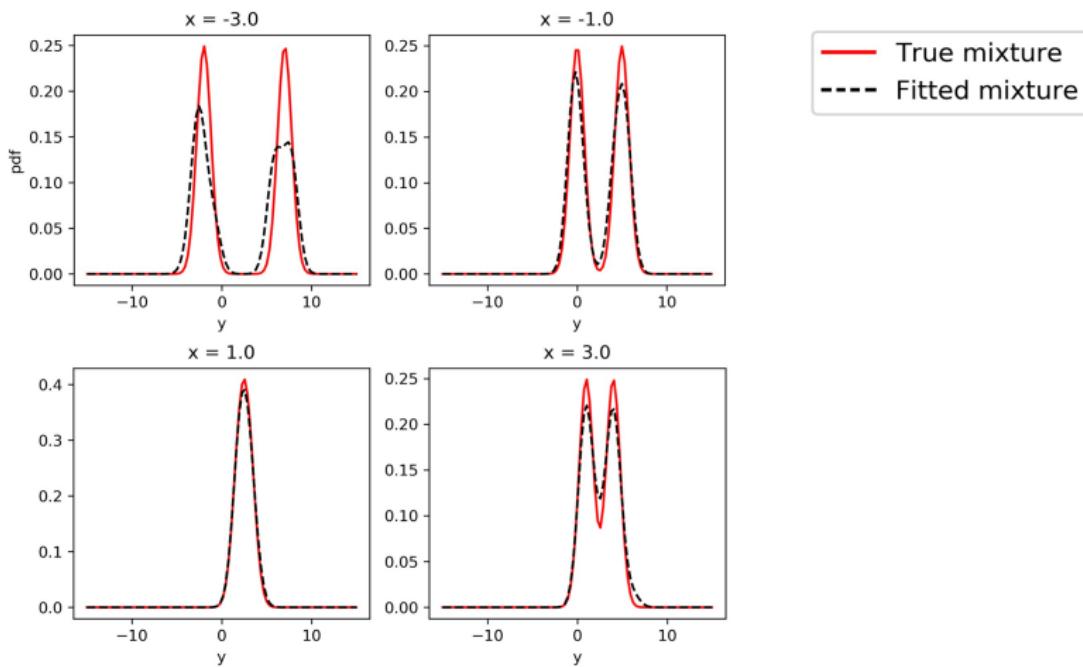


Figure: True and fitted probability density functions (pdf) of  $y$  at different  $x$ 's

## Numerical Example 2

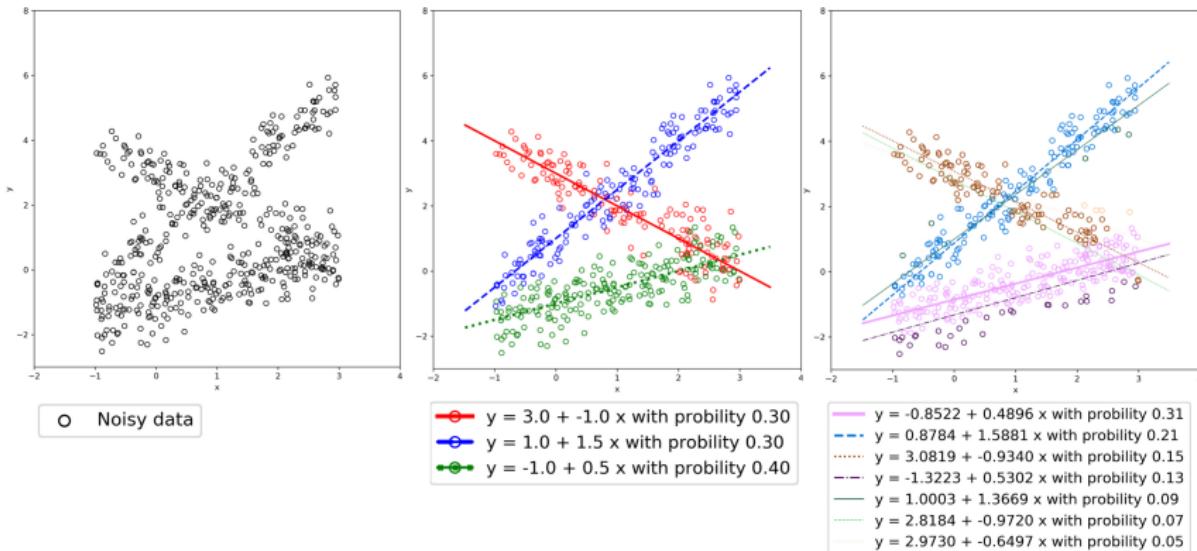


Figure: Left: Noisy data; Middle: True mixture; Right: Fitted mixture

## Numerical Example 2 (Continued)

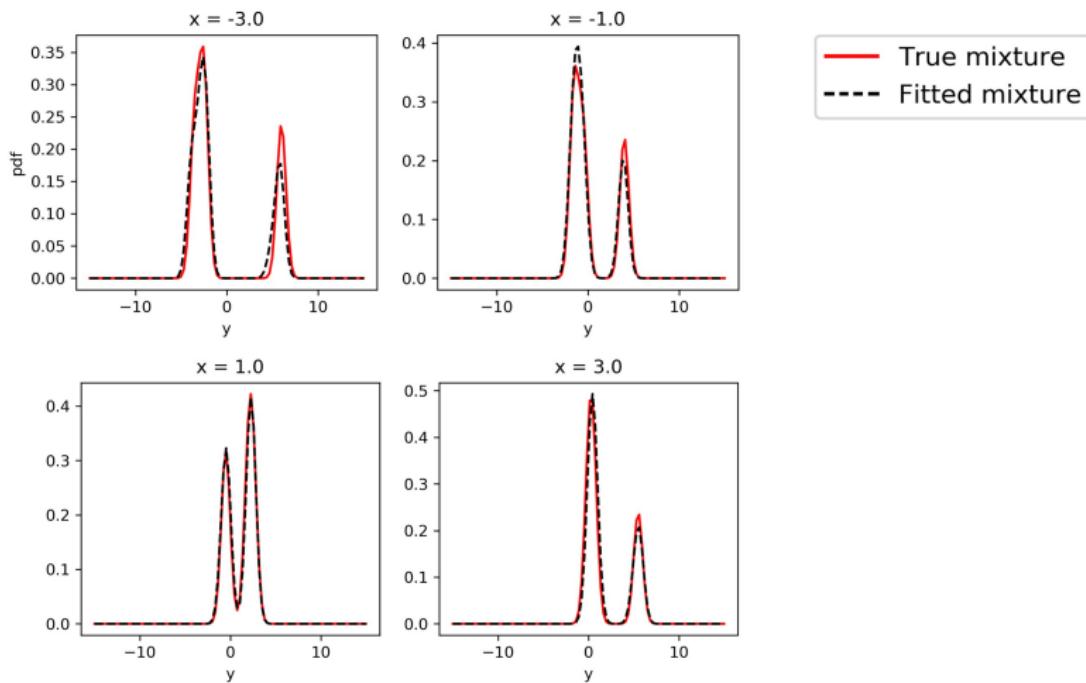


Figure: True and fitted probability density functions (pdf) of  $y$  at different  $x$ 's

# Numerical Example 3

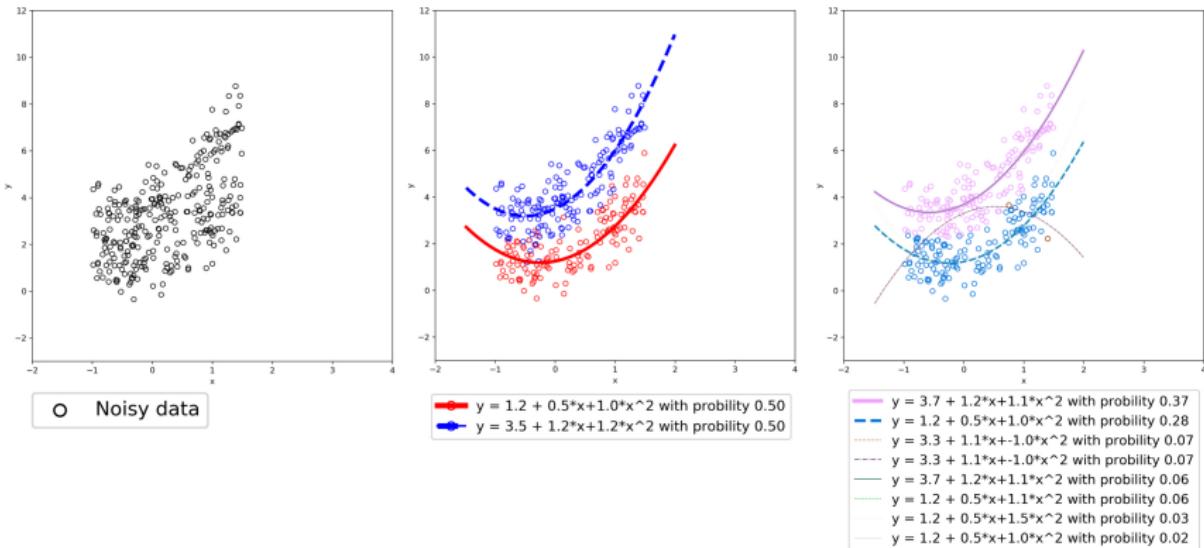


Figure: Left: Noisy data; Middle: True mixture; Right: Fitted mixture

## Numerical Example 3 (Continued)

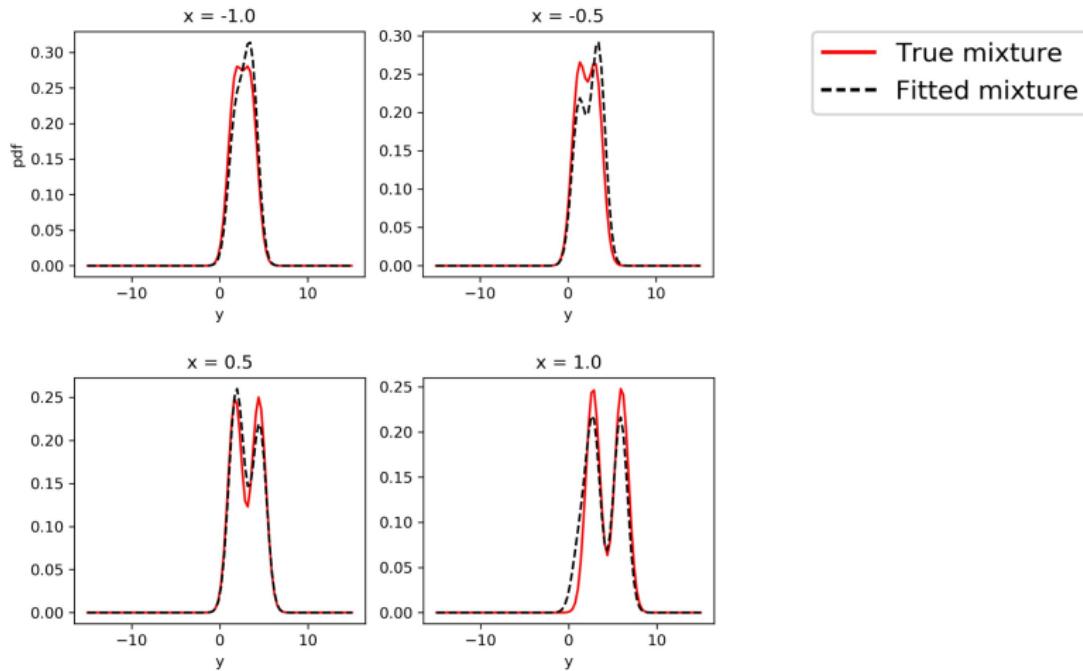


Figure: True and fitted probability density functions (pdf) of  $y$  at different  $x$ 's

# Contents

1 Introduction

2 Fitting MLR with NPMLE

3 Existence and Computation

4 Finite-sample Hellinger Error Bound

5 Summary

## Loss Function

Recall that the conditional density of  $Y$  given  $x$  is  $f_x^{G^*}$  and the estimated conditional density is  $f_x^{\hat{G}}$

# Loss Function

Recall that the conditional density of  $Y$  given  $x$  is  $f_x^{G^*}$  and the estimated conditional density is  $f_x^{\hat{G}}$

## Definition

The squared **Hellinger distance**  $\mathfrak{H}^2(f_x^{\hat{G}}, f_x^{G^*})$  is used as a measure of error in predicting  $y$  for a fixed covariate value  $x$ , where

$$\mathfrak{H}^2(f_x^{\hat{G}}, f_x^{G^*}) = \int \left\{ (f_x^{\hat{G}}(y))^{1/2} - (f_x^{G^*}(y))^{1/2} \right\}^2 dy$$

# Loss Function

Recall that the conditional density of  $Y$  given  $x$  is  $f_x^{G^*}$  and the estimated conditional density is  $f_x^{\hat{G}}$

## Definition

The squared **Hellinger distance**  $\mathfrak{H}^2(f_x^{\hat{G}}, f_x^{G^*})$  is used as a measure of error in predicting  $y$  for a fixed covariate value  $x$ , where

$$\mathfrak{H}^2(f_x^{\hat{G}}, f_x^{G^*}) = \int \left\{ (f_x^{\hat{G}}(y))^{1/2} - (f_x^{G^*}(y))^{1/2} \right\}^2 dy$$

- **Fixed design** Average over  $x_i, i = 1, \dots, n$ , which leads to the loss function

$$\mathfrak{H}_n^2(f_x^{\hat{G}}, f_x^{G^*}) = \frac{1}{n} \sum_{i=1}^n \mathfrak{H}^2(f_{x_i}^{\hat{G}}, f_{x_i}^{G^*})$$

# Loss Function

Recall that the conditional density of  $Y$  given  $x$  is  $f_x^{G^*}$  and the estimated conditional density is  $f_x^{\hat{G}}$

## Definition

The squared **Hellinger distance**  $\mathfrak{H}^2(f_x^{\hat{G}}, f_x^{G^*})$  is used as a measure of error in predicting  $y$  for a fixed covariate value  $x$ , where

$$\mathfrak{H}^2(f_x^{\hat{G}}, f_x^{G^*}) = \int \left\{ (f_x^{\hat{G}}(y))^{1/2} - (f_x^{G^*}(y))^{1/2} \right\}^2 dy$$

- **Fixed design** Average over  $x_i, i = 1, \dots, n$ , which leads to the loss function

$$\mathfrak{H}_n^2(f_x^{\hat{G}}, f_x^{G^*}) = \frac{1}{n} \sum_{i=1}^n \mathfrak{H}^2(f_{x_i}^{\hat{G}}, f_{x_i}^{G^*})$$

- This presentation only covers **fixed design**. Please see our paper for random design

# Finite-sample Bound: Fixed Design

## Theorem

Assume that

- (i)  $\max_{1 \leq i \leq n} \|x_i\| \leq B$
- (ii)  $G^*$  is supported on a ball centered at the origin with radius  $R > 1$

Then

$$E \mathfrak{H}_n^2(f^{\hat{G}}, f^{G^*}) \leq C(p, B, R, \sigma) \frac{(\log n)^{p+1}}{n}$$

# Finite-sample Bound: Fixed Design

## Theorem

Assume that

- (i)  $\max_{1 \leq i \leq n} \|x_i\| \leq B$
- (ii)  $G^*$  is supported on a ball centered at the origin with radius  $R > 1$

Then

$$E \mathfrak{H}_n^2(f^{\hat{G}}, f^{G^*}) \leq C(p, B, R, \sigma) \frac{(\log n)^{p+1}}{n}$$

- When  $p$  is small, one gets nearly the parametric rate

# Finite-sample Bound: Fixed Design

## Theorem

Assume that

- (i)  $\max_{1 \leq i \leq n} \|x_i\| \leq B$
- (ii)  $G^*$  is supported on a ball centered at the origin with radius  $R > 1$

Then

$$E \mathfrak{H}_n^2(f^{\hat{G}}, f^{G^*}) \leq C(p, B, R, \sigma) \frac{(\log n)^{p+1}}{n}$$

- When  $p$  is small, one gets nearly the parametric rate
- Our paper gives an explicit expression for  $C(p, B, R, \sigma)$

# Contents

1 Introduction

2 Fitting MLR with NPMLE

3 Existence and Computation

4 Finite-sample Hellinger Error Bound

5 Summary

# Summary

- We propose to fit mixture of linear regressions with the nonparametric maximum likelihood estimators

## Summary

- We propose to fit mixture of linear regressions with the nonparametric maximum likelihood estimators
- We provide **both algorithmic computing procedures and detailed theoretical analysis**

# Summary

- We propose to fit mixture of linear regressions with the nonparametric maximum likelihood estimators
- We provide **both algorithmic computing procedures and detailed theoretical analysis**
- Our finite-sample bounds for the Hellinger error are **parametric** (up to logarithmic multiplicative factors)

# Summary

- We propose to fit mixture of linear regressions with the nonparametric maximum likelihood estimators
- We provide **both algorithmic computing procedures and detailed theoretical analysis**
- Our finite-sample bounds for the Hellinger error are **parametric** (up to logarithmic multiplicative factors)
- **Future directions**
  - ▶ Other sorts of regression models, such as multivariate linear regression, generalized linear model, and logistic regression
  - ▶ When  $p$  is comparable to  $n$ , some sparsity assumptions might be needed

# Summary

- We propose to fit mixture of linear regressions with the nonparametric maximum likelihood estimators
- We provide **both algorithmic computing procedures and detailed theoretical analysis**
- Our finite-sample bounds for the Hellinger error are **parametric** (up to logarithmic multiplicative factors)
- **Future directions**
  - ▶ Other sorts of regression models, such as multivariate linear regression, generalized linear model, and logistic regression
  - ▶ When  $p$  is comparable to  $n$ , some sparsity assumptions might be needed

**Thank You**  
Any questions or comments?

## References I

- Rudi Beran and P Warwick Millar. Minimum distance estimation in random coefficient regression models. *The Annals of Statistics*, 22(4):1976–1992, 1994.
- Rudolf Beran and Peter Hall. Estimating coefficient distributions in random coefficient regressions. *The Annals of Statistics*, 20(4):1970–1984, 1992.
- Rudolf Beran, Andrey Feuerverger, and Peter Hall. On nonparametric estimation of intercept and slope distributions in random coefficient regression. *The Annals of Statistics*, 24(6):2569–2592, 1996.
- Partha Deb and Ann M Holmes. Estimates of use and costs of behavioural health care: a comparison of standard and finite mixture models. *Health economics*, 9(6):475–489, 2000.
- Marguerite Frank and Philip Wolfe. An algorithm for quadratic programming. *Naval research logistics quarterly*, 3(1-2):95–110, 1956.

## References II

- Stefan Hoderlein, Jussi Klemelä, and Enno Mammen. Analyzing the random coefficient model nonparametrically. *Econometric Theory*, 26(3):804–837, 2010.
- Martin Jaggi. Revisiting frank-wolfe: Projection-free sparse convex optimization. In *Proceedings of The 30th International Conference on Machine Learning*, volume 28, pages 427–435. Curran, 2013.
- Michael I Jordan and Robert A Jacobs. Hierarchical mixtures of experts and the em algorithm. *Neural computation*, 6(2):181–214, 1994.
- Tze Leung Lai and Mei-Chiung Shih. Nonparametric estimation in nonlinear mixed effects models. *Biometrika*, 90(1):1–13, 2003.
- Yuanzhi Li and Yingyu Liang. Learning mixtures of linear regressions with nearly optimal complexity. *arXiv preprint arXiv:1802.07895*, 2018.
- Bruce G Lindsay. The geometry of mixture likelihoods: a general theory. *The Annals of Statistics*, pages 86–94, 1983.

## References III

Richard E Quandt. The estimation of the parameters of a linear regression system obeying two separate regimes. *Journal of the american statistical association*, 53(284):873–880, 1958.

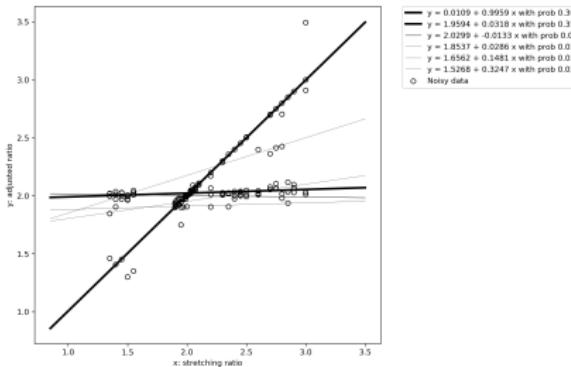
Michel Wedel and Wagner A Kamakura. *Market segmentation: Conceptual and methodological foundations*, volume 8. Springer Science & Business Media, 2012.

Chien-Fu Wu. Some algorithmic aspects of the theory of optimal designs. *The Annals of Statistics*, pages 1286–1301, 1978.

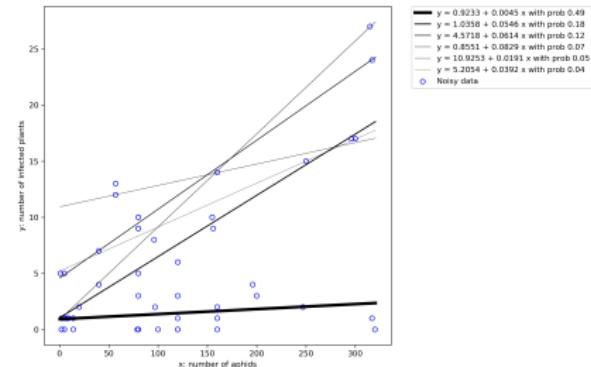
Xinyang Yi and Constantine Caramanis. Regularized em algorithms: A unified framework and statistical guarantees. In *Advances in Neural Information Processing Systems*, pages 1567–1575, 2015.



# Illustration on Real Data



(a) Music perception data



(b) Aphids data

Figure: Real data experiments

# Finite-sample Bound: Random Design

## Theorem

$$\begin{aligned} & \int \mathfrak{H}(f^{\tilde{G}}, f^{G^*}) d\mu(x) \\ & \leq \left( \frac{C_p}{\min(1 - \alpha_1, \alpha_2)} \right)^{1/2} \epsilon_n + \frac{\rho(\mathfrak{L}_{S_0}, R, p)}{n^{1/2}} + \frac{2(\log n)^{1/2}}{n^{1/2}} \end{aligned}$$

with probability at least  $1 - 3n^{-1}$ , where

$$\epsilon_n^2 = \left( 1 + \frac{2R\mathfrak{L}_{S_0}}{\sigma\sqrt{2\log(3n^2)}} \right)^p \frac{(\log n)^{p+1}}{n}$$

# Finite-sample Bound: Random Design

## Theorem

$$\begin{aligned} & \int \mathfrak{H}(f^{\tilde{G}}, f^{G^*}) d\mu(x) \\ & \leq \left( \frac{C_p}{\min(1 - \alpha_1, \alpha_2)} \right)^{1/2} \epsilon_n + \frac{\rho(\mathfrak{L}_{S_0}, R, p)}{n^{1/2}} + \frac{2(\log n)^{1/2}}{n^{1/2}} \end{aligned}$$

with probability at least  $1 - 3n^{-1}$ , where

$$\epsilon_n^2 = \left( 1 + \frac{2R\mathfrak{L}_{S_0}}{\sigma \sqrt{2 \log(3n^2)}} \right)^p \frac{(\log n)^{p+1}}{n}$$

## Theorem

Under certain assumptions,

$$d(\hat{G}_n, G^*) \rightarrow 0 \text{ in probability}$$