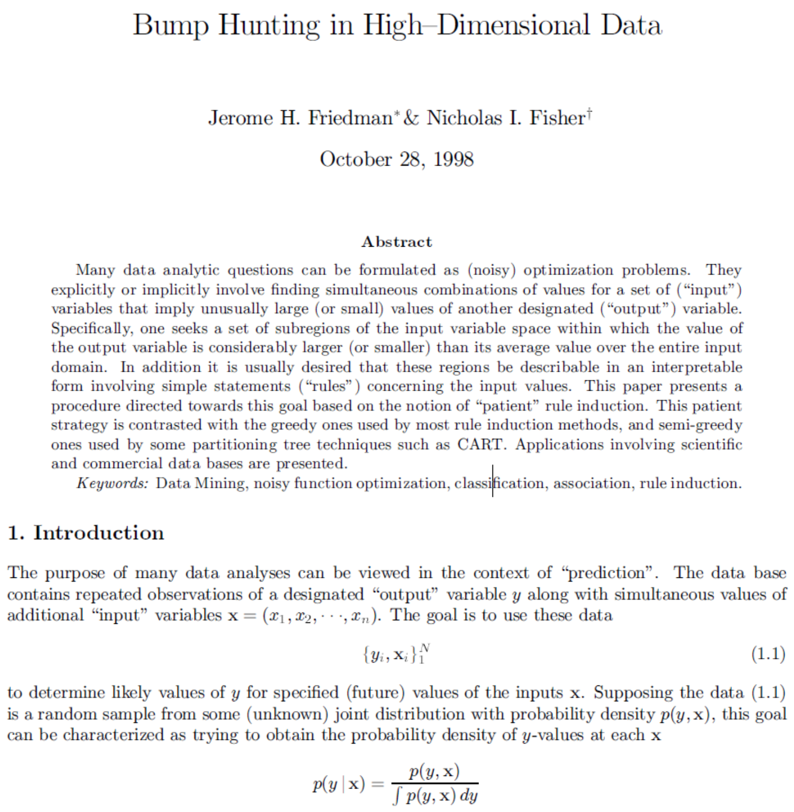
Bump Hunting

(no kidding, that’s the technical term)

Group Meeting 2014-06-10

BLFoley

Papers being referenced here



Bump Hunting in High–Dimensional Data

Jerome H. Friedman[*] & Nicholas I. Fisher[†]

October 28, 1998

Abstract

Many data analytic questions can be formulated as (noisy) optimization problems. They explicitly or implicitly involve finding simultaneous combinations of values for a set of ("input") variables that imply unusually large (or small) values of another designated ("output") variable. Specifically, one seeks a set of subregions of the input variable space within which the value of the output variable is considerably larger (or smaller) than its average value over the entire input domain. In addition it is usually desired that these regions be describable in an interpretable form involving simple statements ("rules") concerning the input values. This paper presents a procedure directed towards this goal based on the notion of "patient" rule induction. This patient strategy is contrasted with the greedy ones used by most rule induction methods, and semi-greedy ones used by some partitioning tree techniques such as CART. Applications involving scientific and commercial data bases are presented.

Keywords: Data Mining, noisy function optimization, classification, association, rule induction.

1. Introduction

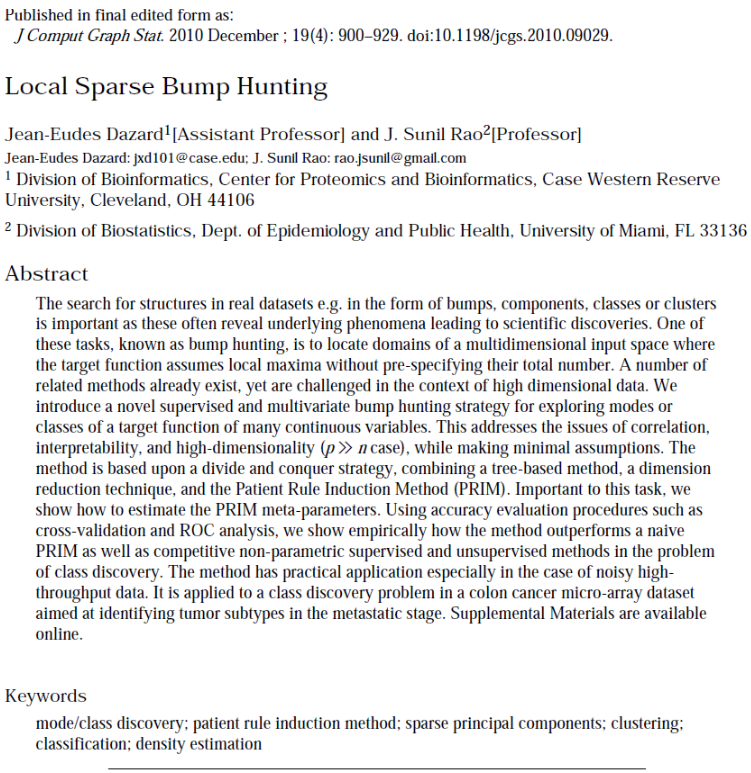
The purpose of many data analyses can be viewed in the context of "prediction". The data base contains repeated observations of a designated "output" variable y along with simultaneous values of additional "input" variables $\mathbf{x} = (x_1, x_2, \cdots, x_n)$. The goal is to use these data

$$\{y_i, \mathbf{x}_i\}_1^N \tag{1.1}$$

to determine likely values of y for specified (future) values of the inputs $\mathbf{x}$. Supposing the data (1.1) is a random sample from some (unknown) joint distribution with probability density $p(y, \mathbf{x})$, this goal can be characterized as trying to obtain the probability density of y-values at each $\mathbf{x}$

$$p(y \mid \mathbf{x}) = \frac{p(y, \mathbf{x})}{\int p(y, \mathbf{x})\, dy}$$

Statistics and Computing. 1999 Dec.; 9(2); 123-143. doi: 10.1023/A:1008894516817



Published in final edited form as:
J Comput Graph Stat. 2010 December ; 19(4): 900–929. doi:10.1198/jcgs.2010.09029.

Local Sparse Bump Hunting

Jean-Eudes Dazard[1][Assistant Professor] and J. Sunil Rao[2][Professor]

Jean-Eudes Dazard: jxd101@case.edu; J. Sunil Rao: rao.jsunil@gmail.com

[1] Division of Bioinformatics, Center for Proteomics and Bioinformatics, Case Western Reserve University, Cleveland, OH 44106

[2] Division of Biostatistics, Dept. of Epidemiology and Public Health, University of Miami, FL 33136

Abstract

The search for structures in real datasets e.g. in the form of bumps, components, classes or clusters is important as these often reveal underlying phenomena leading to scientific discoveries. One of these tasks, known as bump hunting, is to locate domains of a multidimensional input space where the target function assumes local maxima without pre-specifying their total number. A number of related methods already exist, yet are challenged in the context of high dimensional data. We introduce a novel supervised and multivariate bump hunting strategy for exploring modes or classes of a target function of many continuous variables. This addresses the issues of correlation, interpretability, and high-dimensionality ($p \gg n$ case), while making minimal assumptions. The method is based upon a divide and conquer strategy, combining a tree-based method, a dimension reduction technique, and the Patient Rule Induction Method (PRIM). Important to this task, we show how to estimate the PRIM meta-parameters. Using accuracy evaluation procedures such as cross-validation and ROC analysis, we show empirically how the method outperforms a naive PRIM as well as competitive non-parametric supervised and unsupervised methods in the problem of class discovery. The method has practical application especially in the case of noisy high-throughput data. It is applied to a class discovery problem in a colon cancer micro-array dataset aimed at identifying tumor subtypes in the metastatic stage. Supplemental Materials are available online.

Keywords

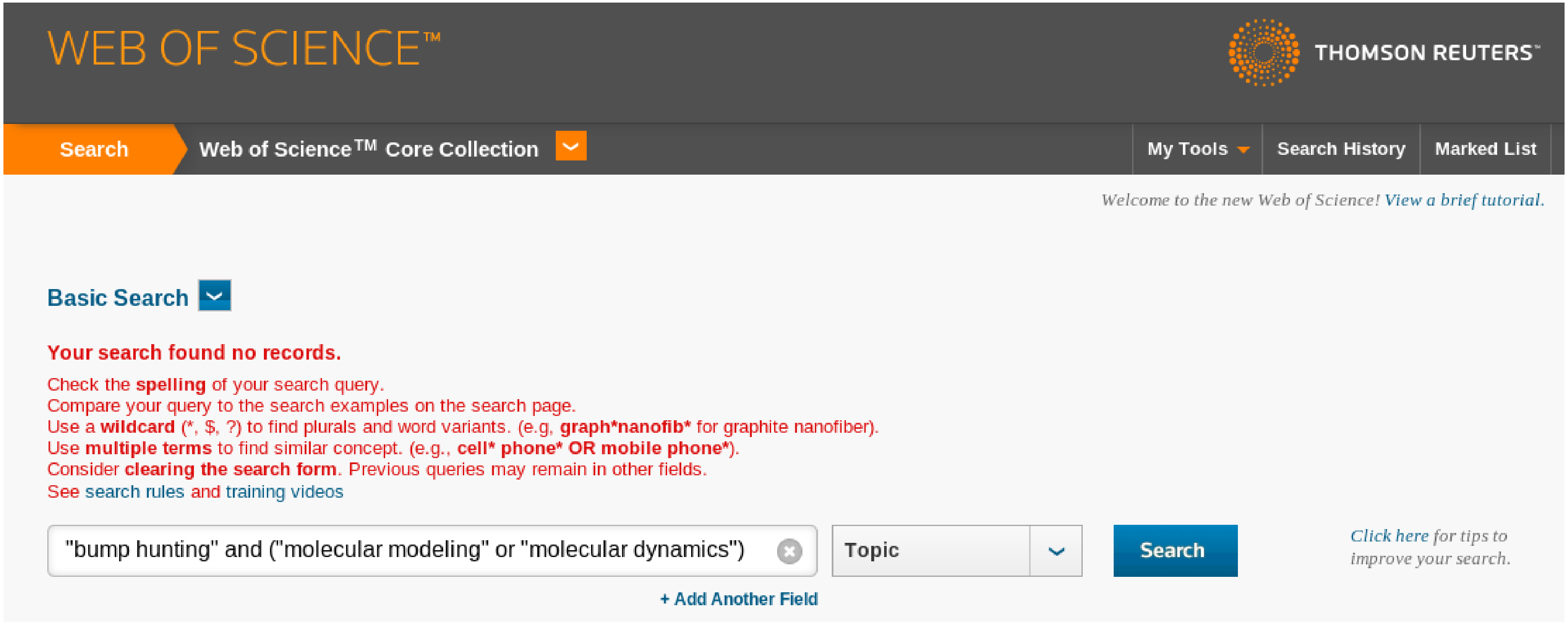
mode/class discovery; patient rule induction method; sparse principal components; clustering; classification; density estimation

J Comput Graph Stat. 2010 Dec.; 19(4): 900–929. doi:10.1198/jcgs.2010.09029

Caveats / Meta-Info

- This method is new to me
- There are many variants and extensions
    - The other paper is one
- The main paper (“Bump Hunting….”):
    - Presents a nice review of previous work
    - Emphasizes the needs of the user
        - Strict adherence to protocol not always required
        - Examples here are per the paper, best I can tell
    - Presents a new way to hunt bumps, PRIM
        - PRIM is available in R
        - Currently installed on Rime
        - I don’t know how to use it
- Hunting maxima here (minima: the opposite)

Not used by many colleagues



Welcome to the new Web of Science! View a brief tutorial.

Basic Search

Your search found no records.

Check the **spelling** of your search query.
Compare your query to the search examples on the search page.
Use a **wildcard** (*, $, ?) to find plurals and word variants. (e.g, **graph*nanofib*** for graphite nanofiber).
Use **multiple terms** to find similar concept. (e.g., **cell* phone* OR mobile phone***).
Consider **clearing the search form**. Previous queries may remain in other fields.
See search rules and training videos

"bump hunting" and ("molecular modeling" or "molecular dynamics")

Topic

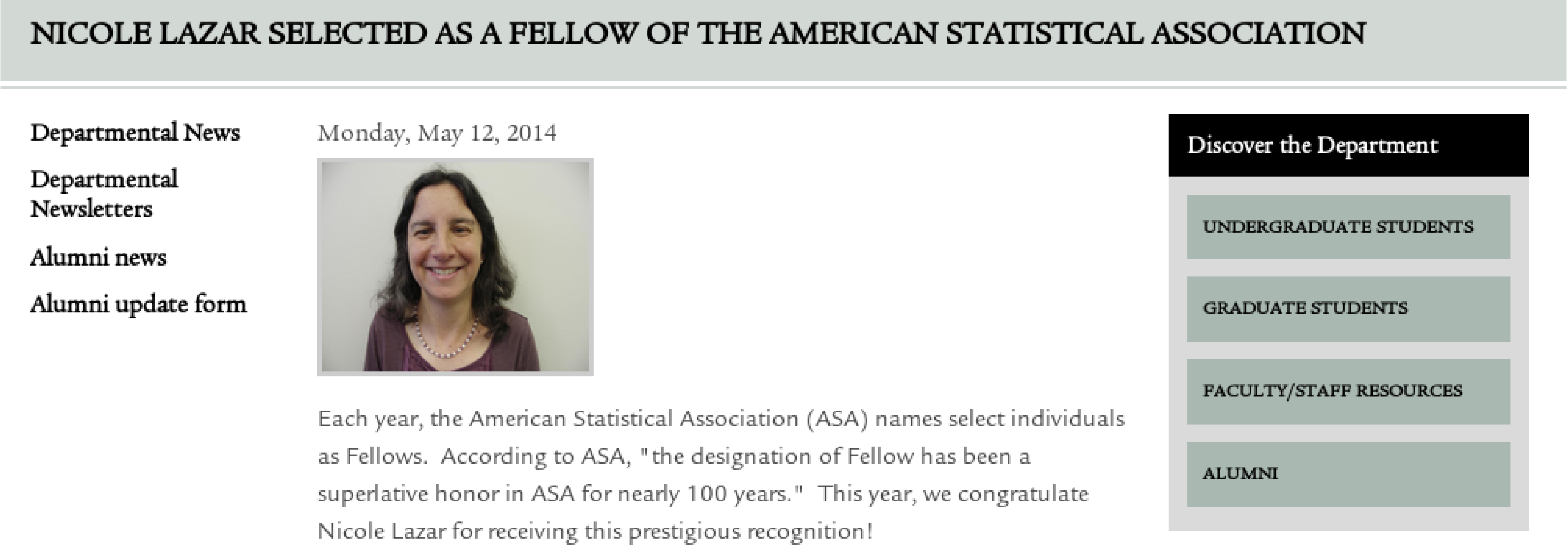
Search

Click here for tips to improve your search.

+ Add Another Field

Provenance

- I’ve been giving sucrose data to Stats students
  - Two years now
  - They work hard, but available methods not so good
- Nicole Lazar, Stats Prof
  - Interested, also works with neuro-imaging data
  - Told me about this method / main paper



NICOLE LAZAR SELECTED AS A FELLOW OF THE AMERICAN STATISTICAL ASSOCIATION

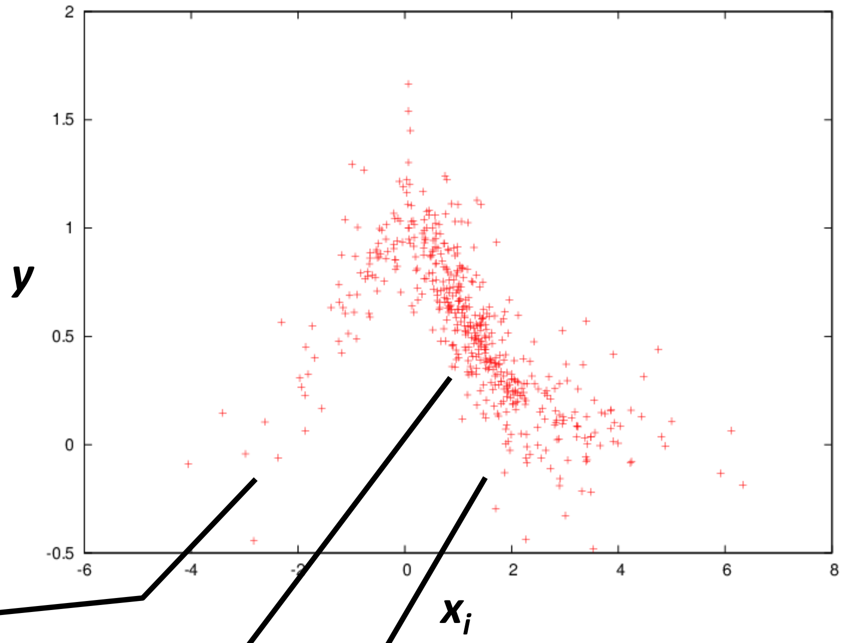
Monday, May 12, 2014

Each year, the American Statistical Association (ASA) names select individuals as Fellows. According to ASA, "the designation of Fellow has been a superlative honor in ASA for nearly 100 years." This year, we congratulate Nicole Lazar for receiving this prestigious recognition!

Why Hunt Bumps?

Consider a scenario like:

- You have data that
  - Approximates some function
  - Possibly not evenly
  - And noisily
- But, you want to find
  - All maxima
  - Possibly even the shape



This is probably a function

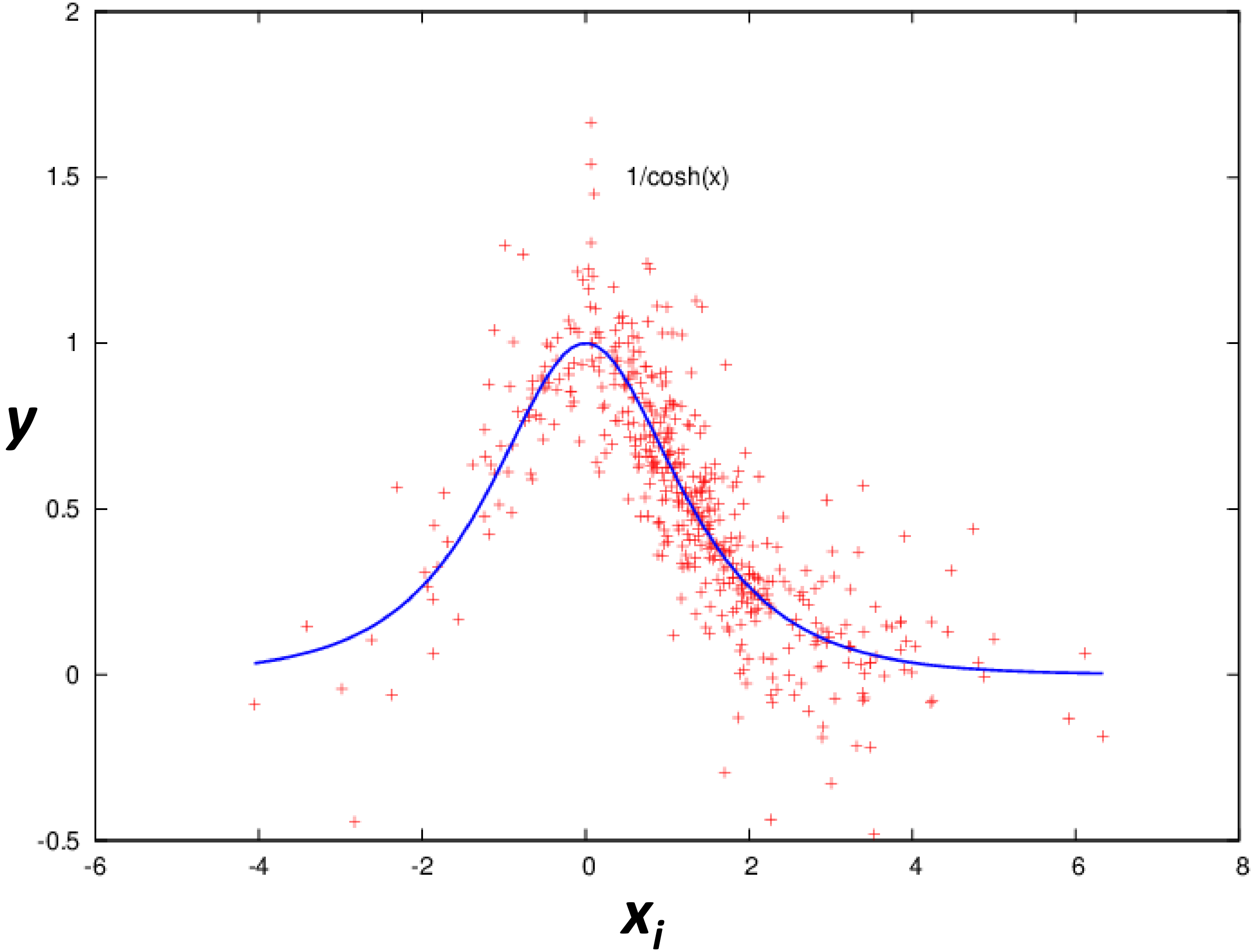
The majority of points might not appear at the function's apparent maximum

The data are very noisy

(btw, I made this data up)

By the way, here is the "real" function

- Of course, in reality you rarely know this



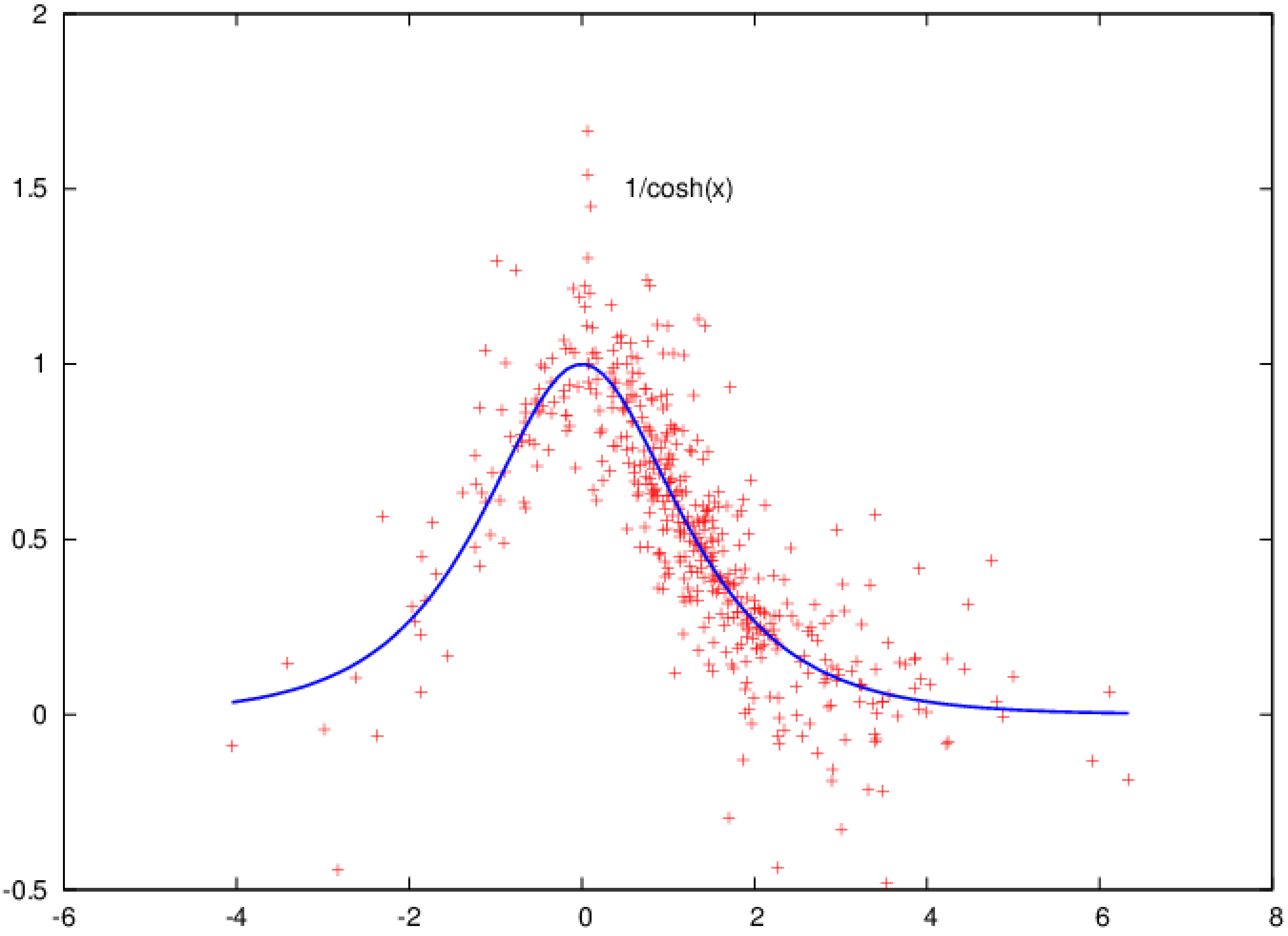
How it works, Part 1

- “Peel” data along each axis, x_i:
  - **Remove fractions**, α, **of the data**
    - Not like histogramming!
  - Remove from one end or the other of the axis
  - **Choose the end that leaves the largest average in *y***
  - Repeat **until only the fraction, β_0, of the data is left**

Example 1

Step 1: Choose values for α and β.

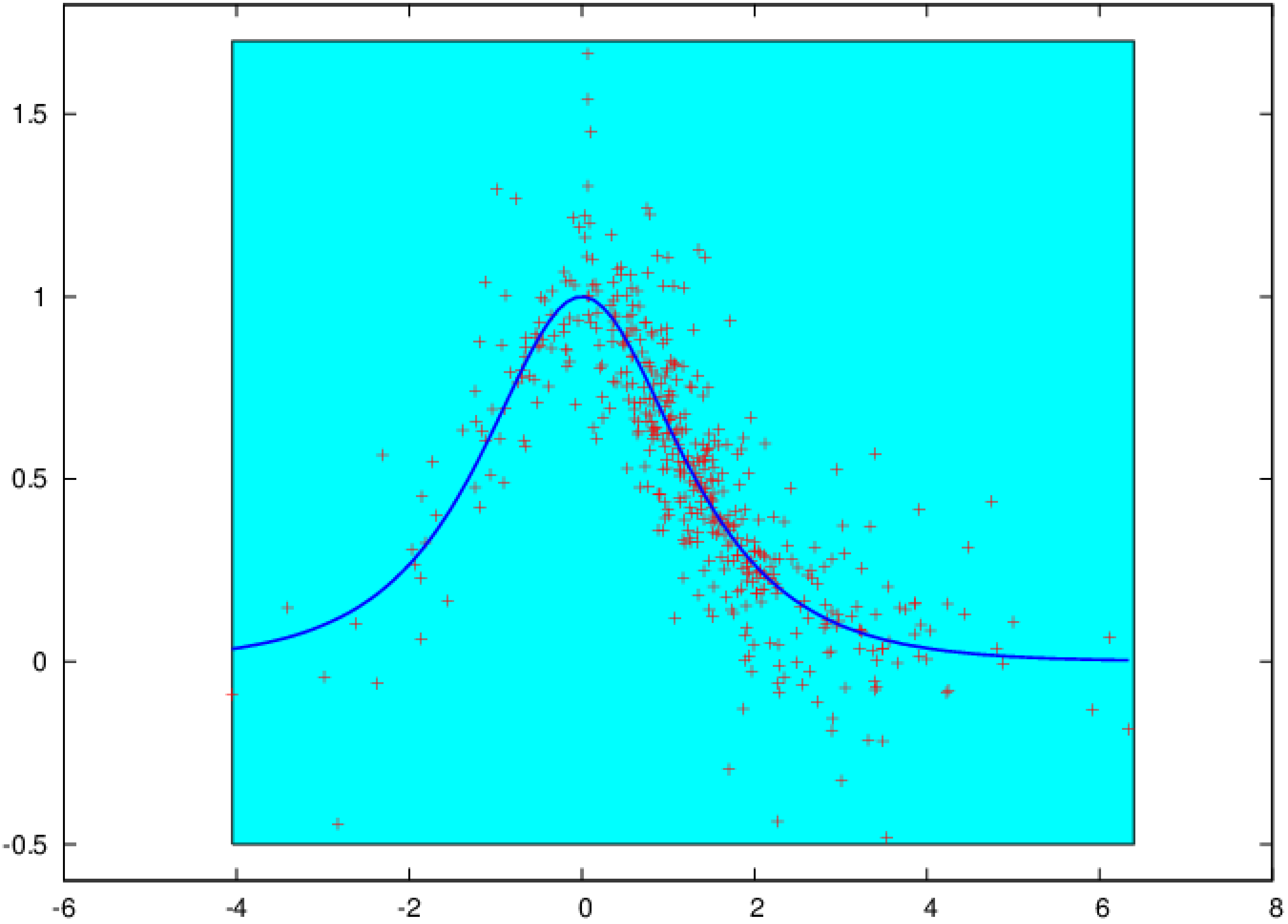
Here: α = 0.1 (50 points) β = 0.2 (100 points) *Note point density; N=500*



Example 1

Step 2: Enclose your data in a “box”.

For this example, the “box” is only along the x axis from ~-4.1 to ~6.5



Example 1

Step 3: Calculate <*y*> if you remove α (=0.1, 50 pts) from:

3a: the high-end of *x*

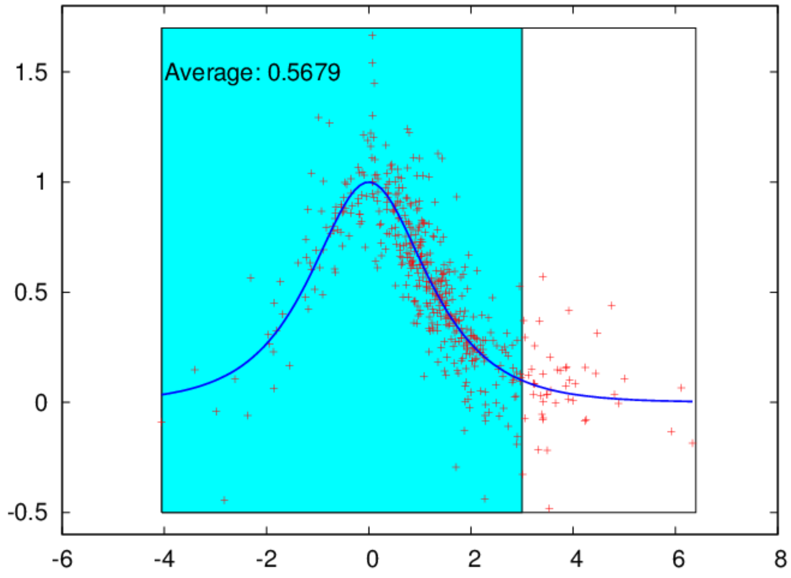
3b: the low end of *x*

Choose **largest** value of <*y*> remaining **(Here: 3a)**

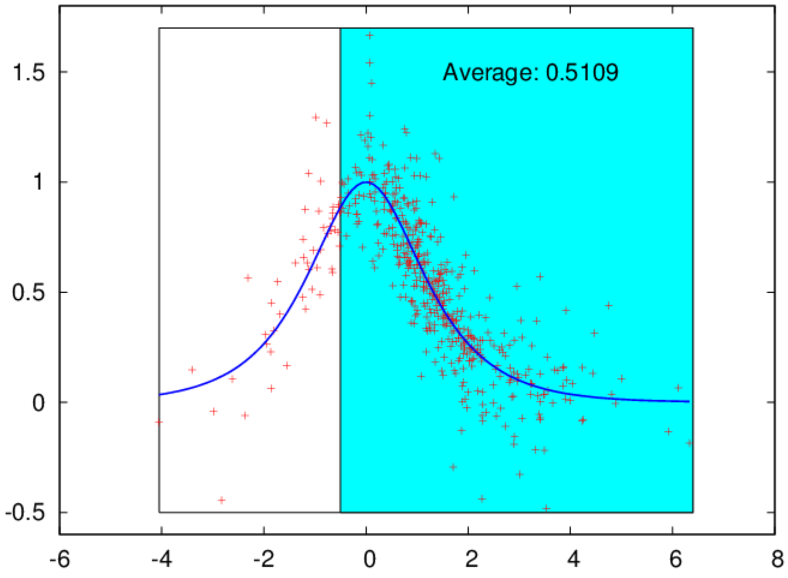
Note: **concern is average *leftover*, not removed**

We also don't care what the original average was (necessarily, yet)

3a: Remove α from top



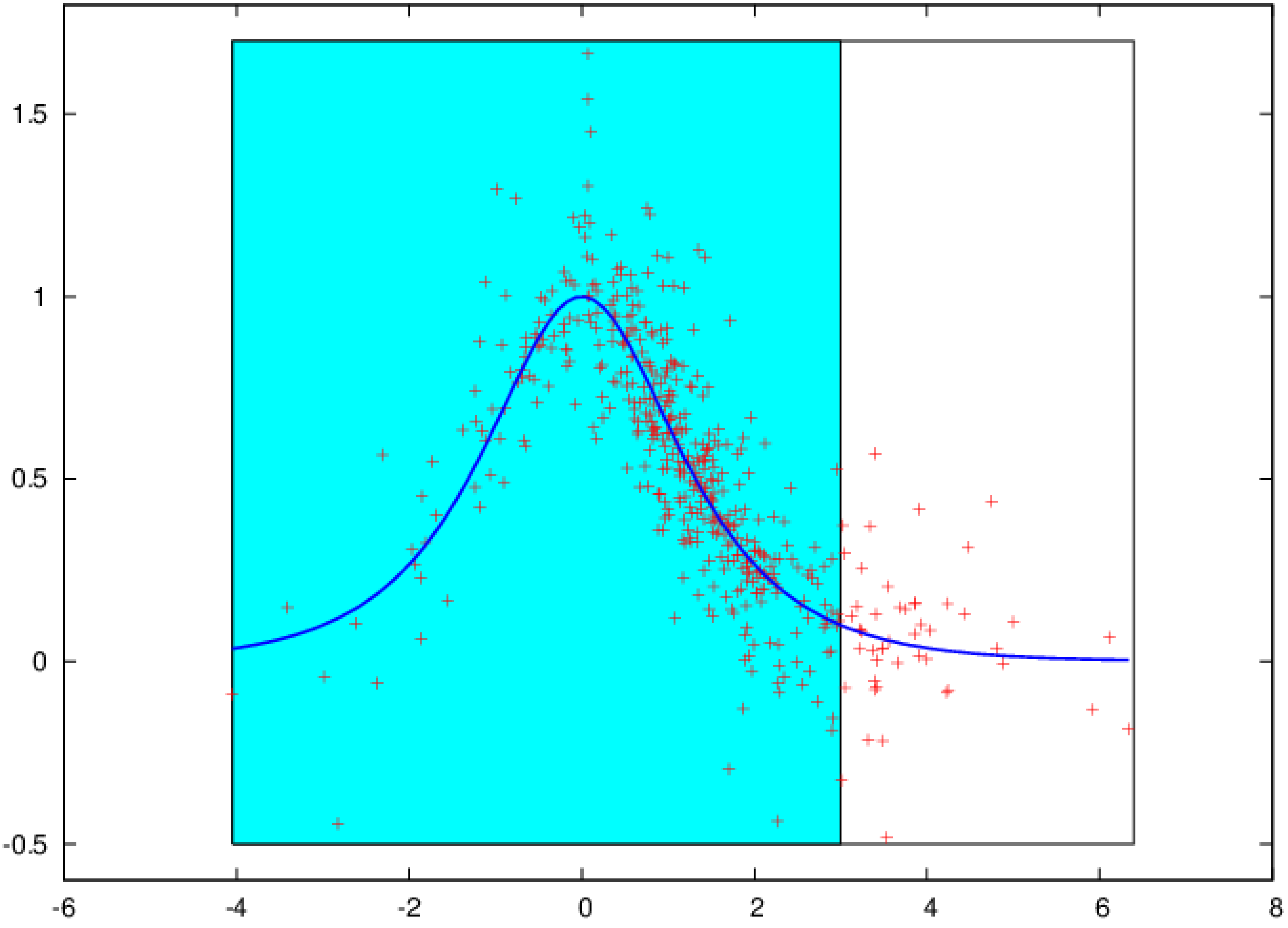
3b: Remove α from bottom



Example 1

Step 4: Remove (peel) those values and repeat

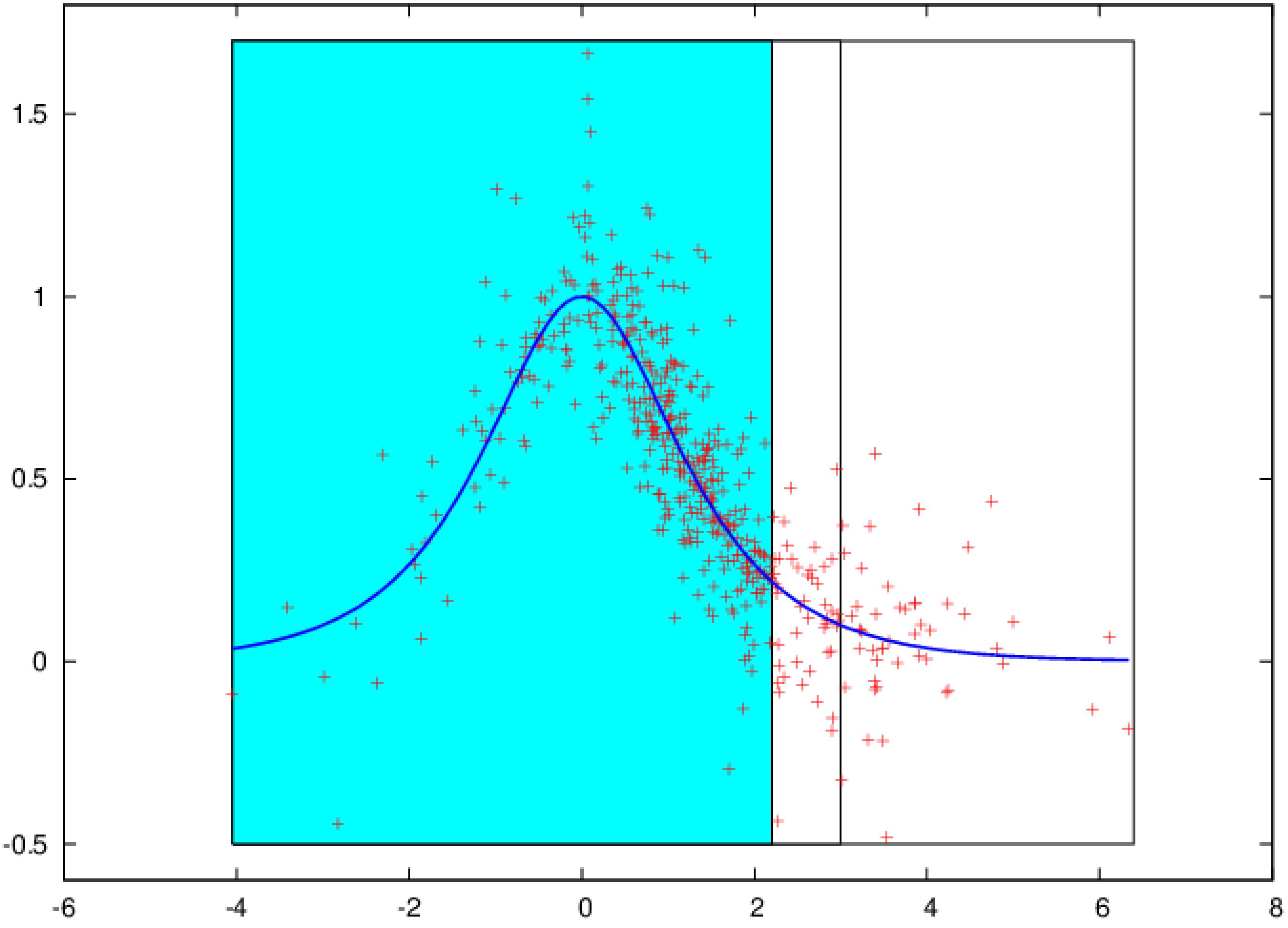
Each image represents removal of 50 points.



Example 1

Step 4: Remove (peel) those values and repeat

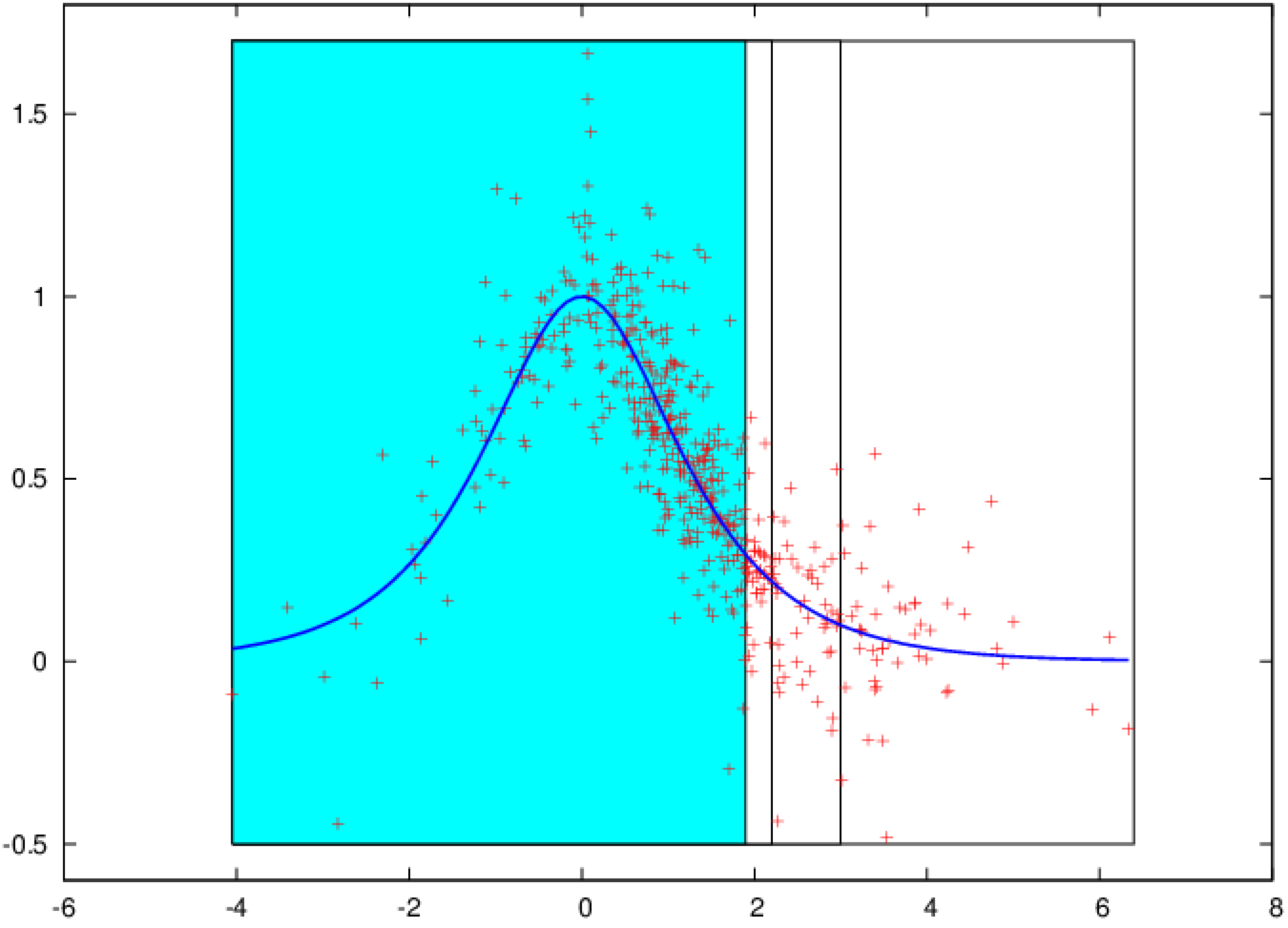
Each image represents removal of 50 points.



Example 1

Step 4: Remove (peel) those values and repeat

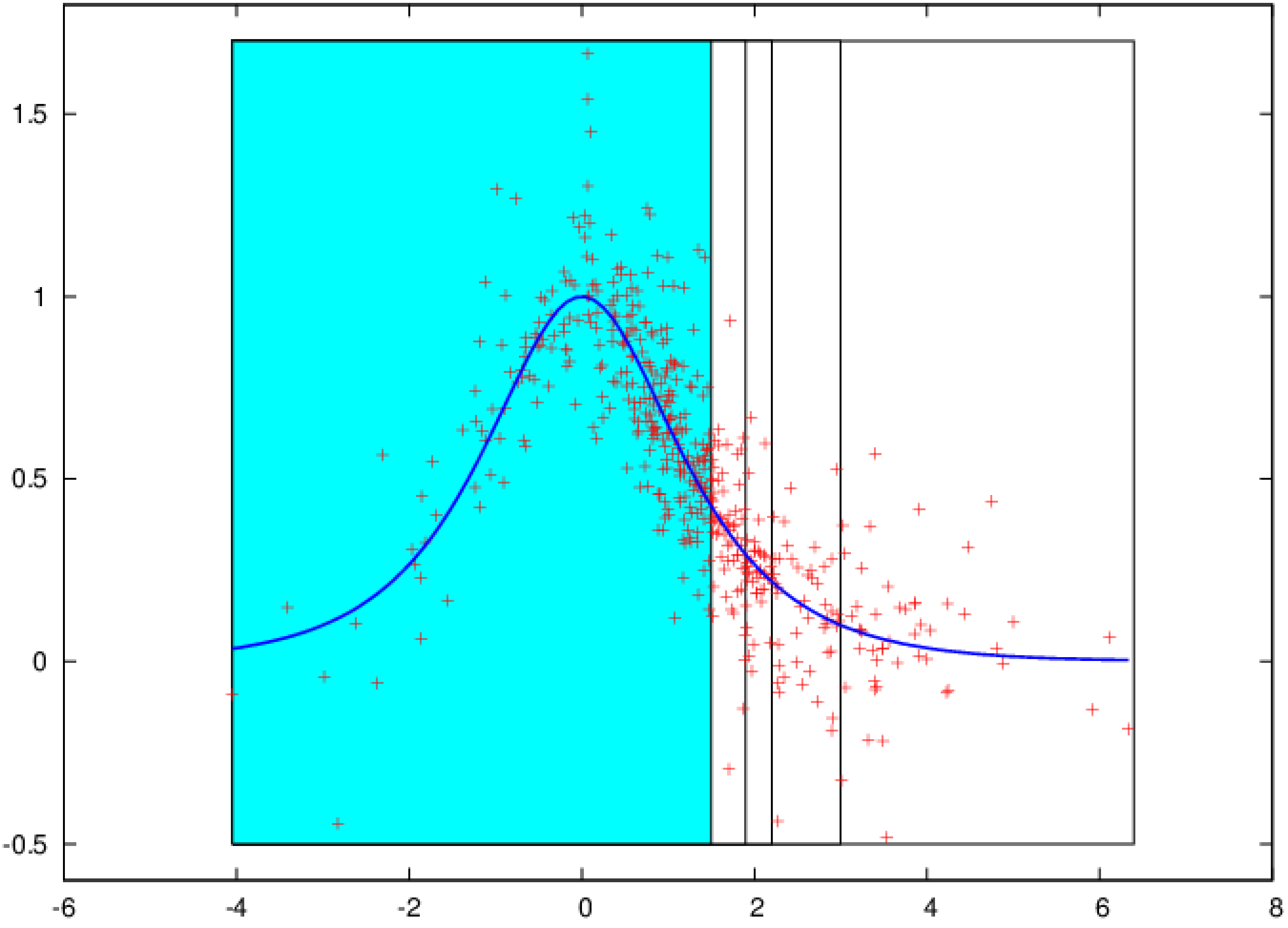
Each image represents removal of 50 points.



Example 1

Step 4: Remove (peel) those values and repeat

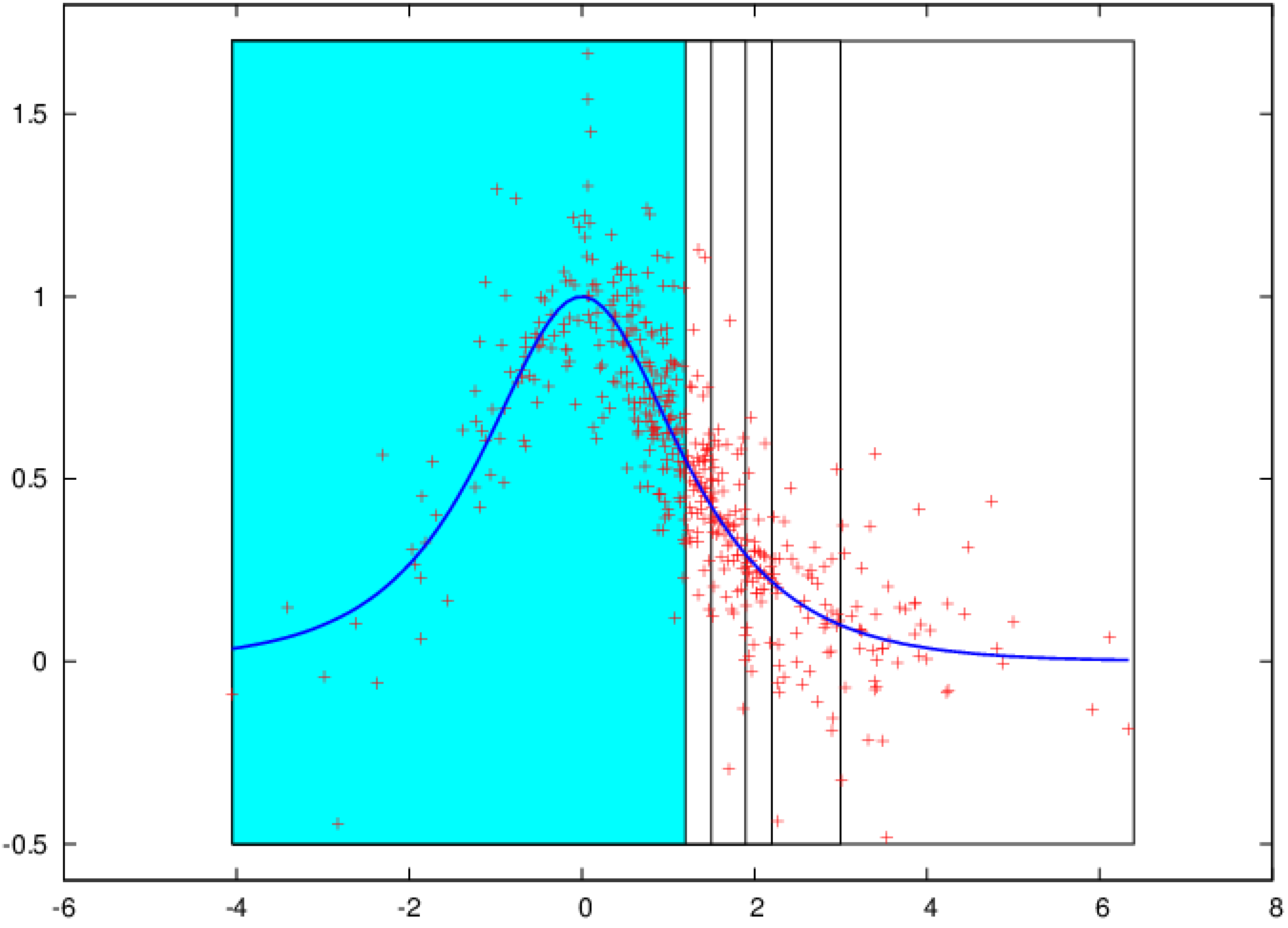
Each image represents removal of 50 points.



Example 1

Step 4: Remove (peel) those values and repeat

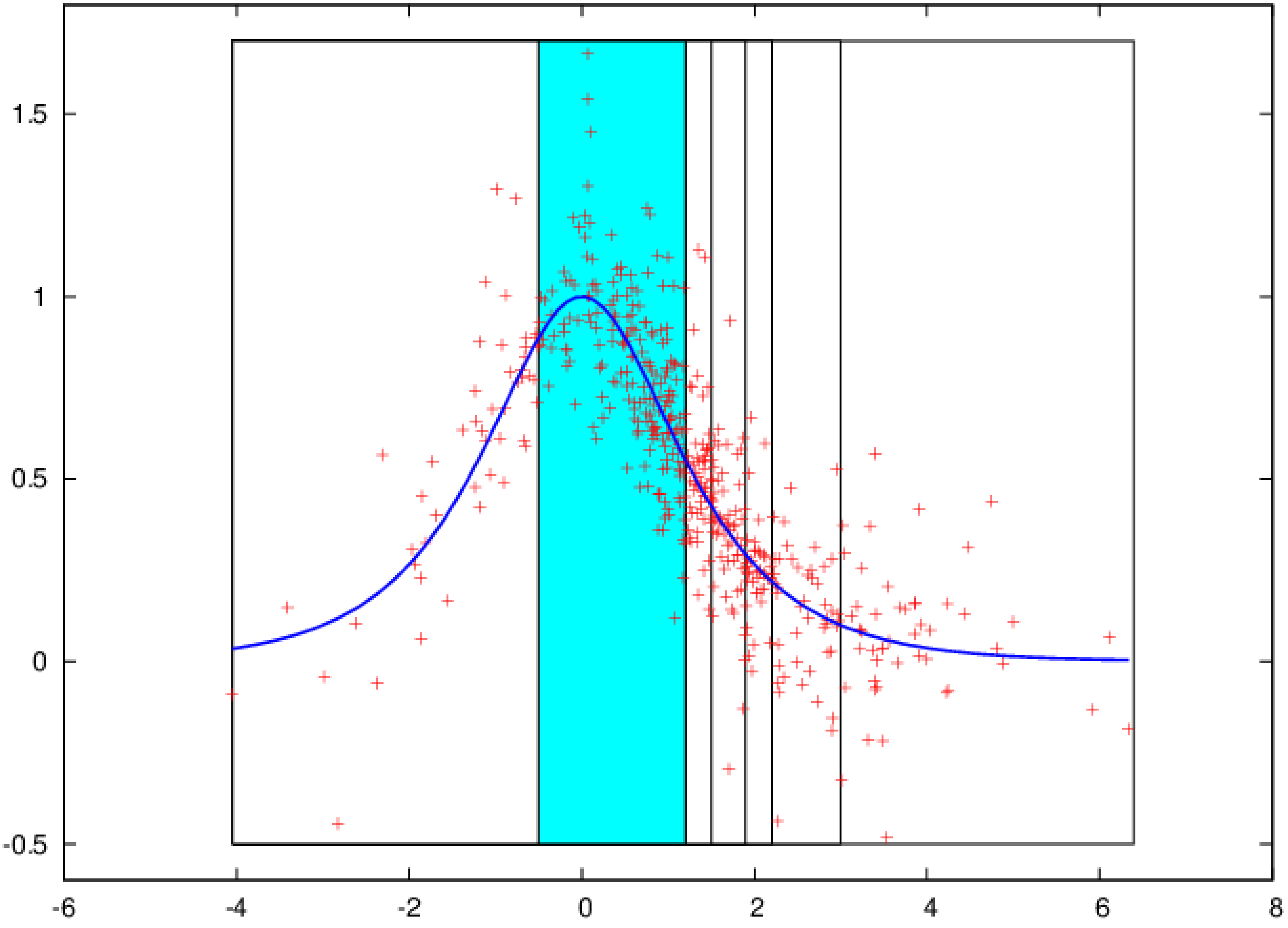
Each image represents removal of 50 points.



Example 1

Step 4: Remove (peel) those values and repeat

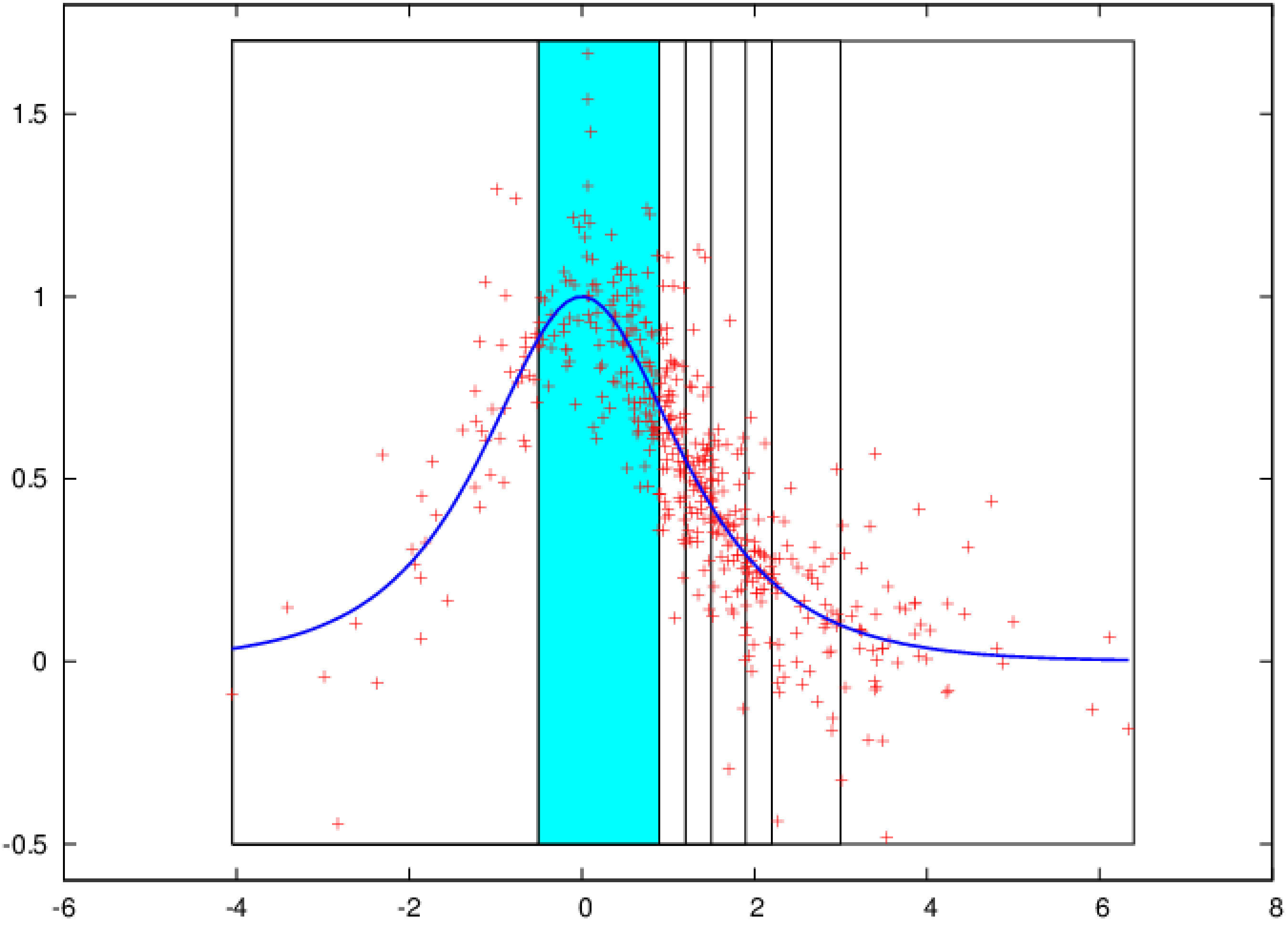
Each image represents removal of 50 points.



Example 1

Step 4: Remove (peel) those values and repeat

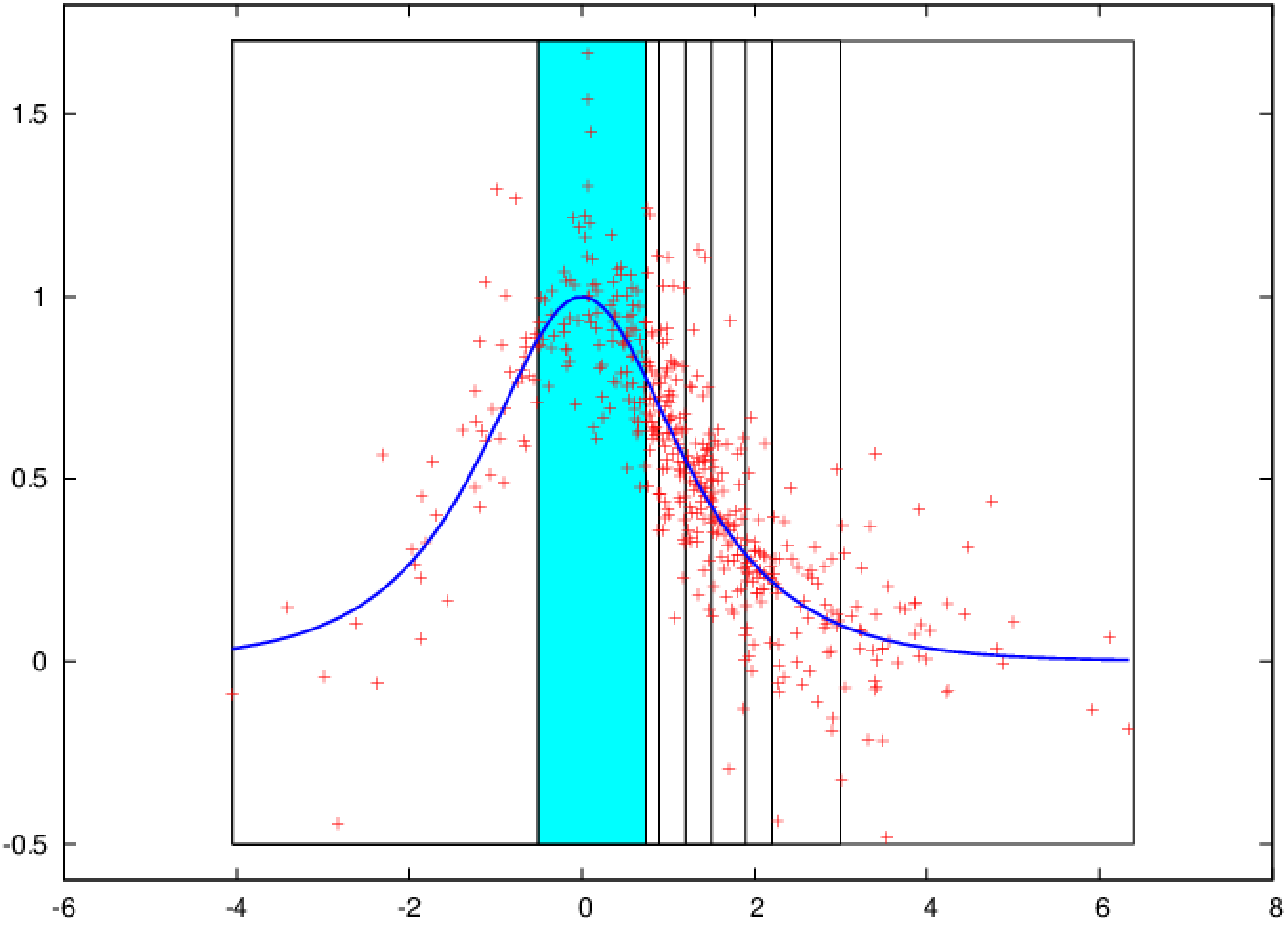
Each image represents removal of 50 points.



Example 1

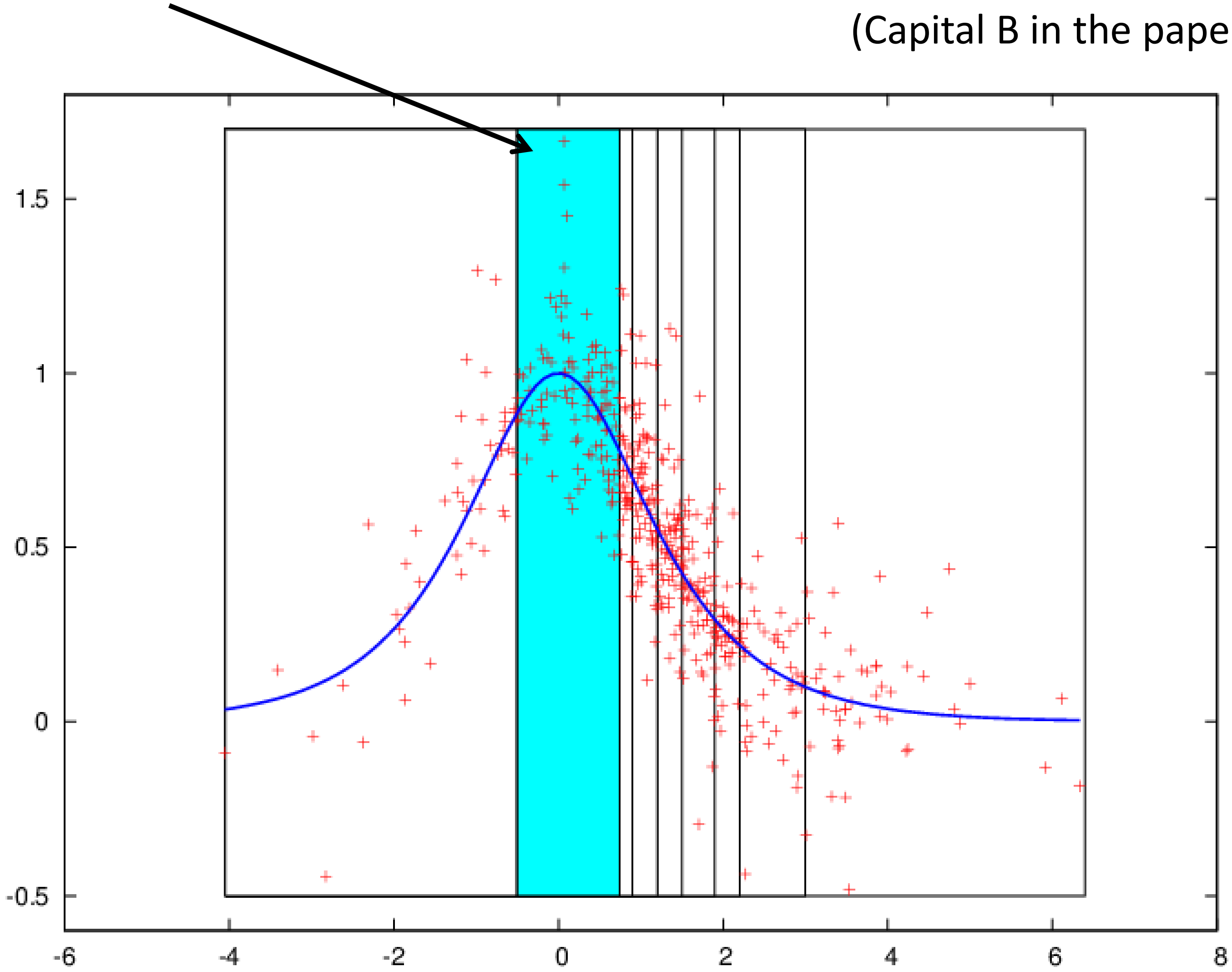
Step 5: **Stop when you reach the target β** (0.2, 100 points)

Here, the current average is 0.9226 $\langle y \rangle$ for all 500: 0.5183



Example 1

Step 6 (optional):
Remove those points from your data set and go hunting for more “bumps”
(Capital B in the paper)



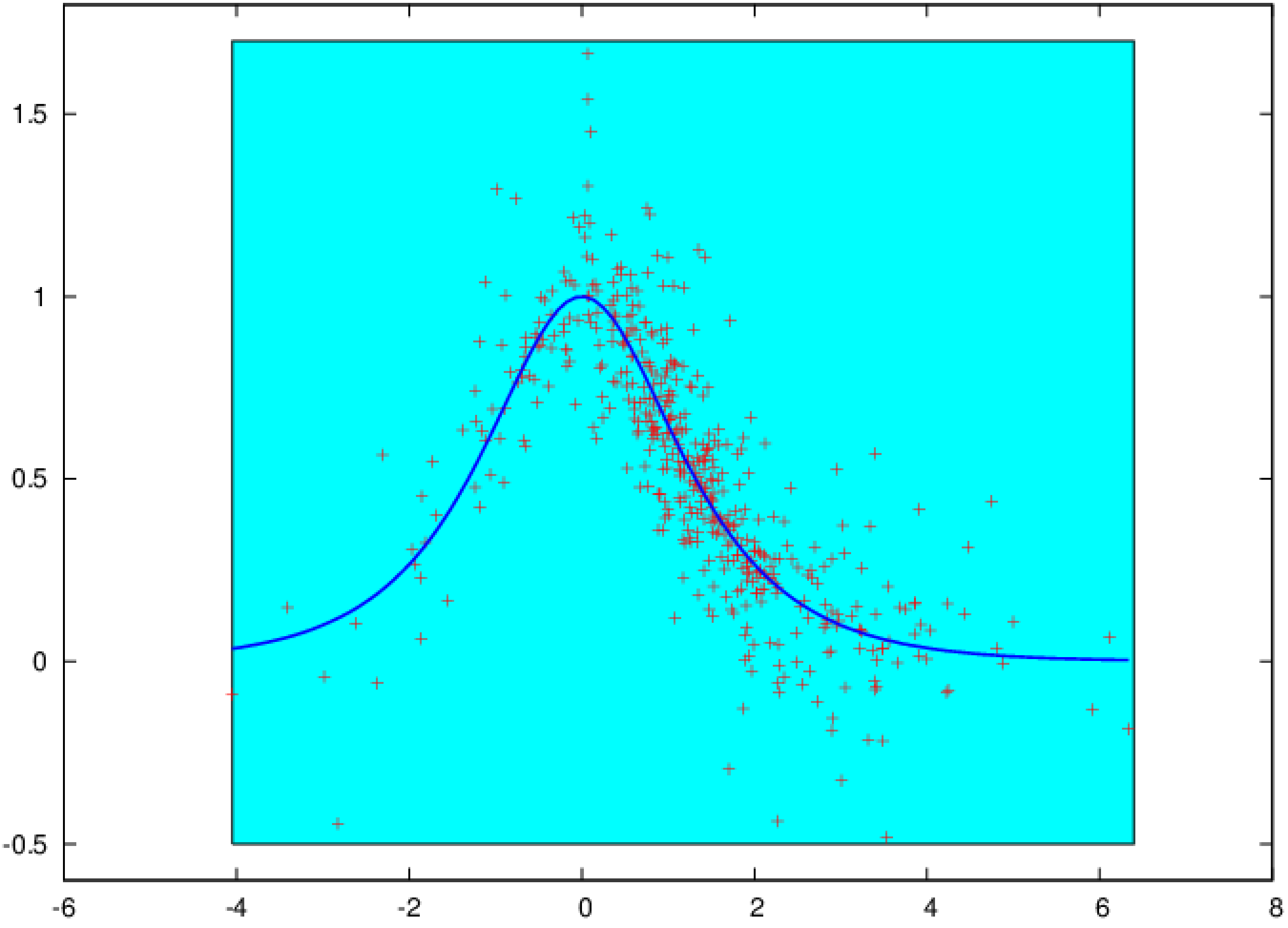
How it works, Part 2

- The value of α matters
  - Small = “Patient” (the P in PRIM)
    - And more computational time
  - Large = “Greedy”
    - But with less-good results
- **“Pasting” is a way to try to fix**
  - Choose a new α
    - Can be the same; sometimes smaller
  - β is not relevant
    - Per the standard protocol
  - Instead, we monitor $<y_i>$
  - ***Add nearby points back in until $<y_i>$ gets smaller***

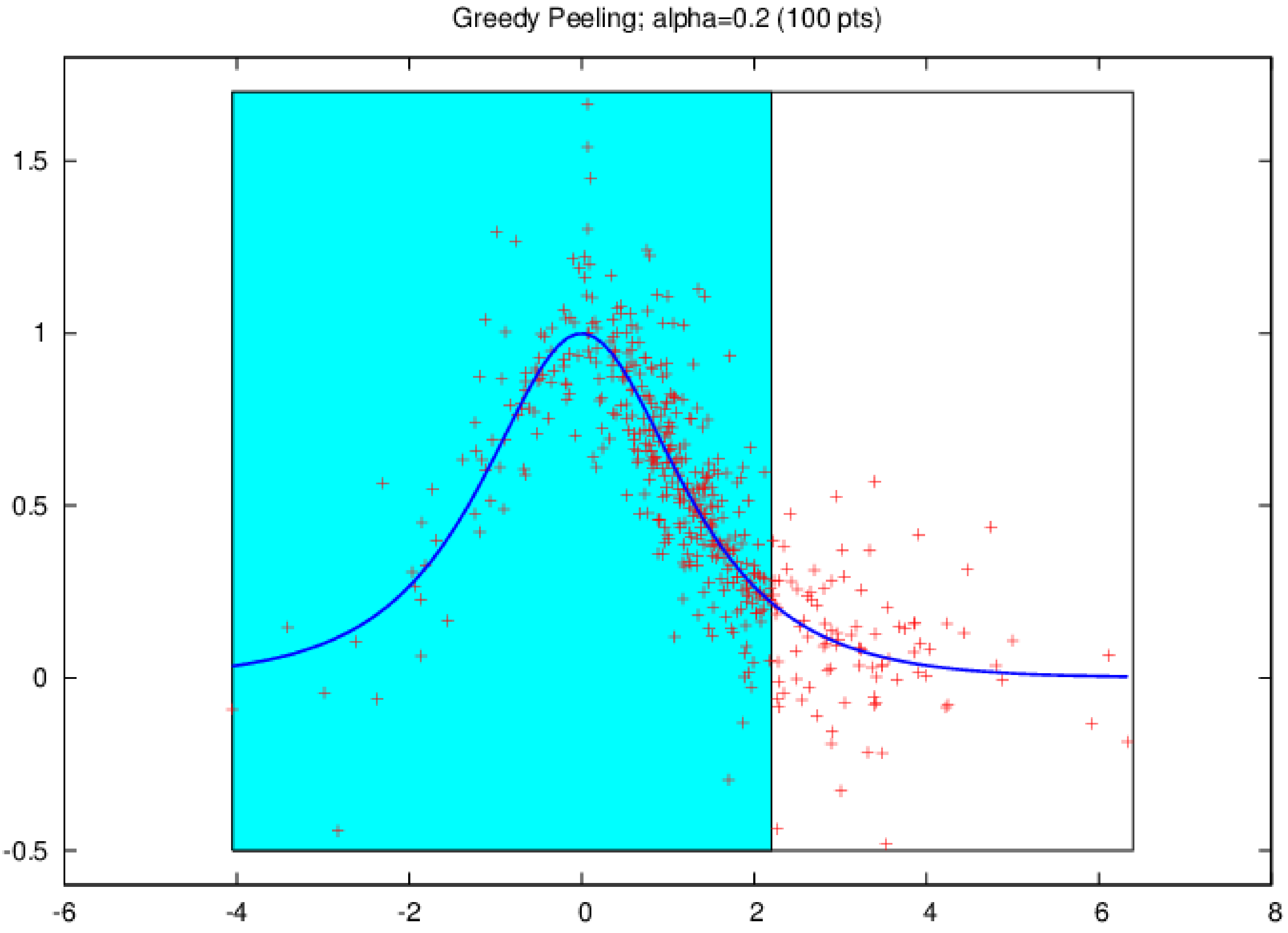
Example 2a

These images show “greedy” peeling, with

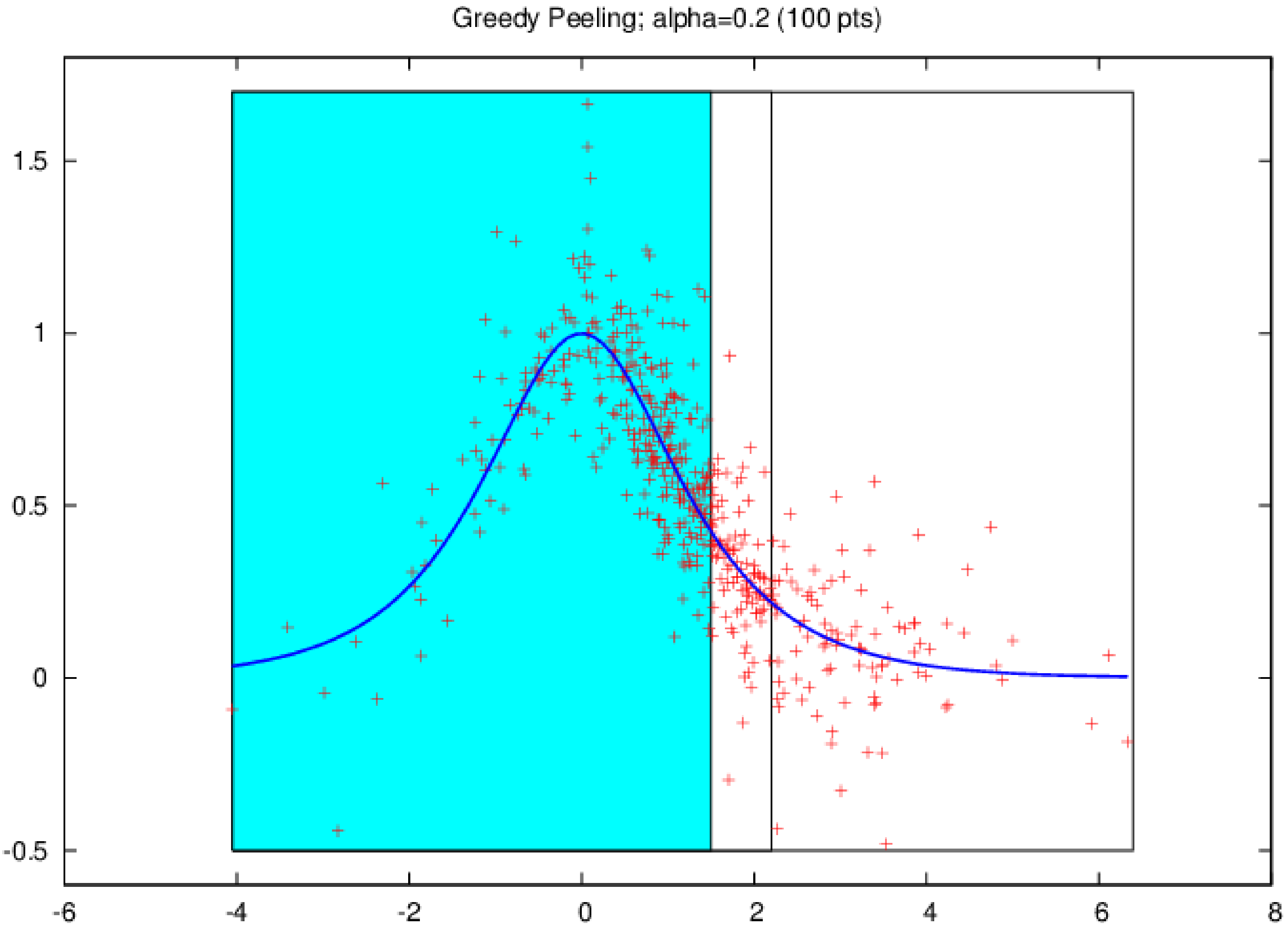
α = 0.2 (100 points) β = 0.2 (100 points)



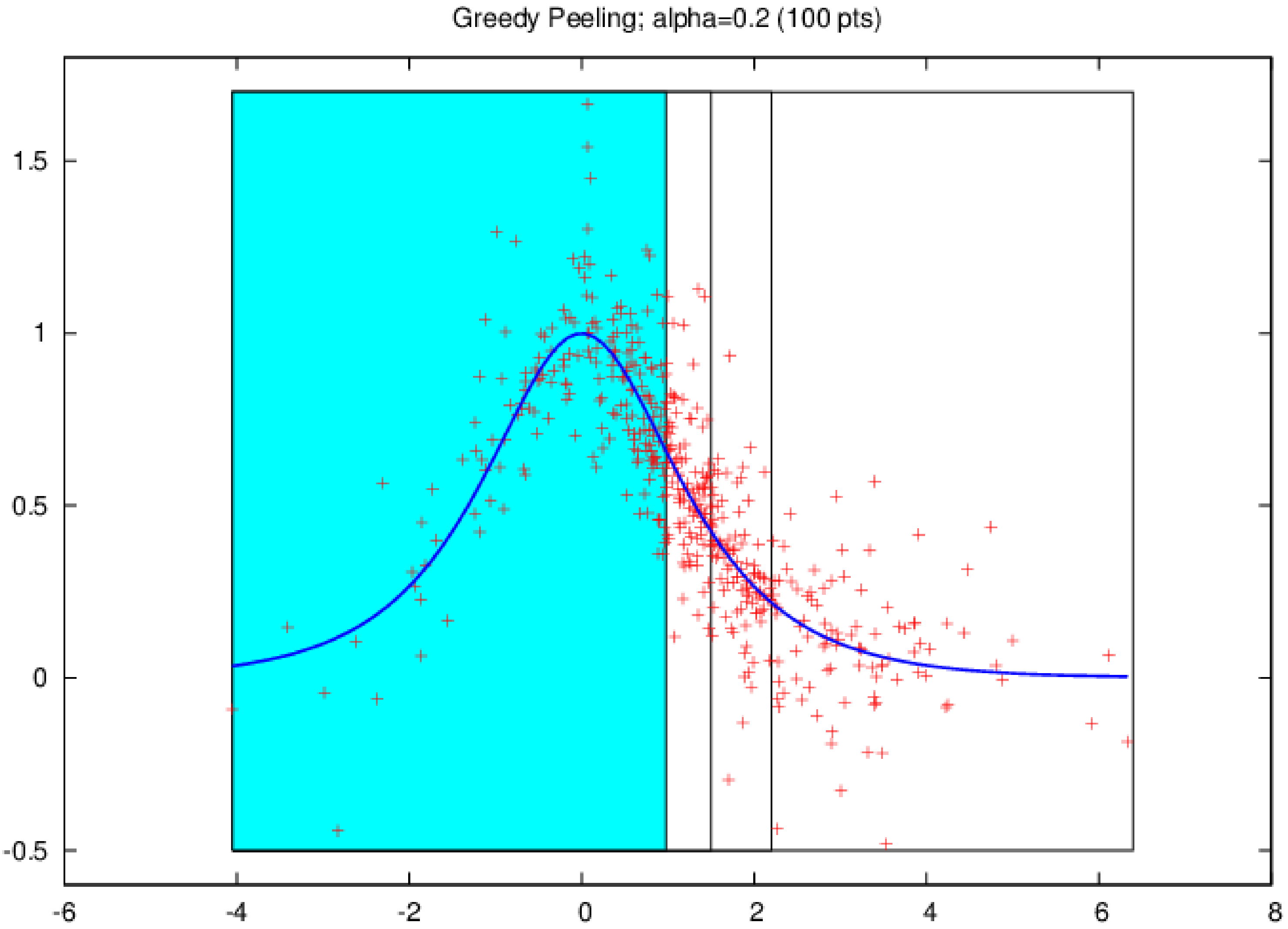
Example 2a



Example 2a

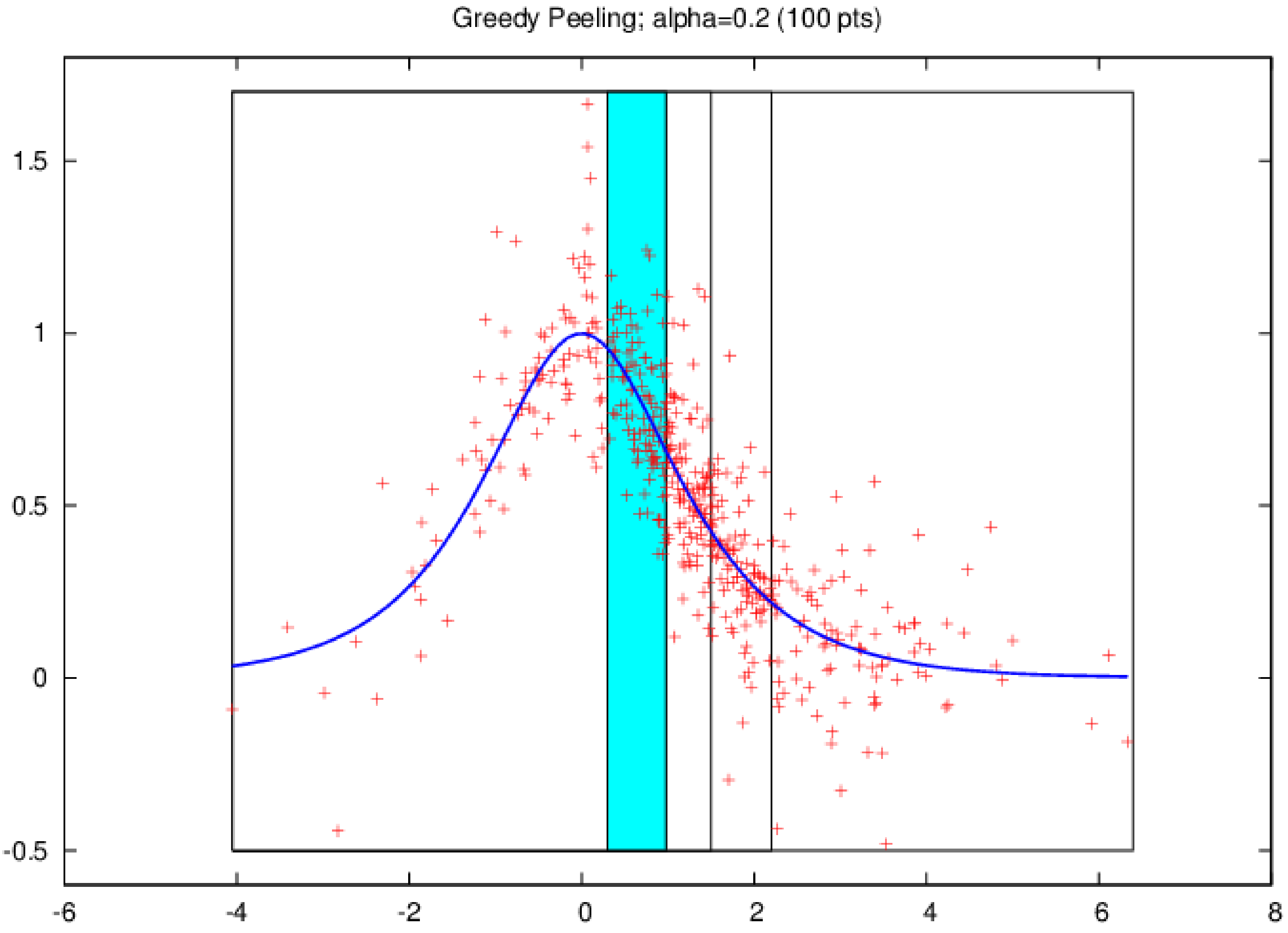


Example 2a



Example 2a

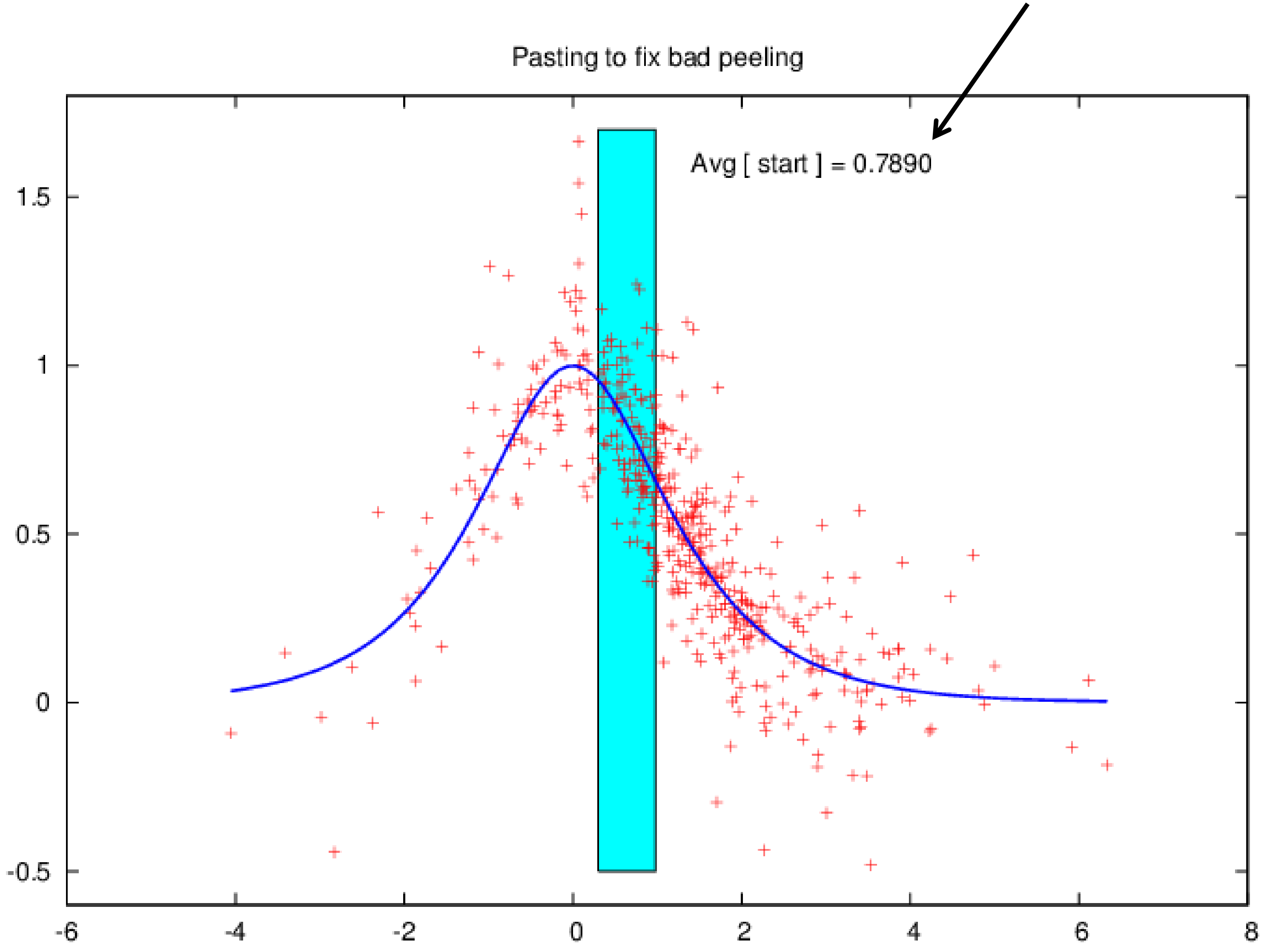
We stop here because **there are only 100 points left** (β = 0.2) . Now, to paste.



Example 2b

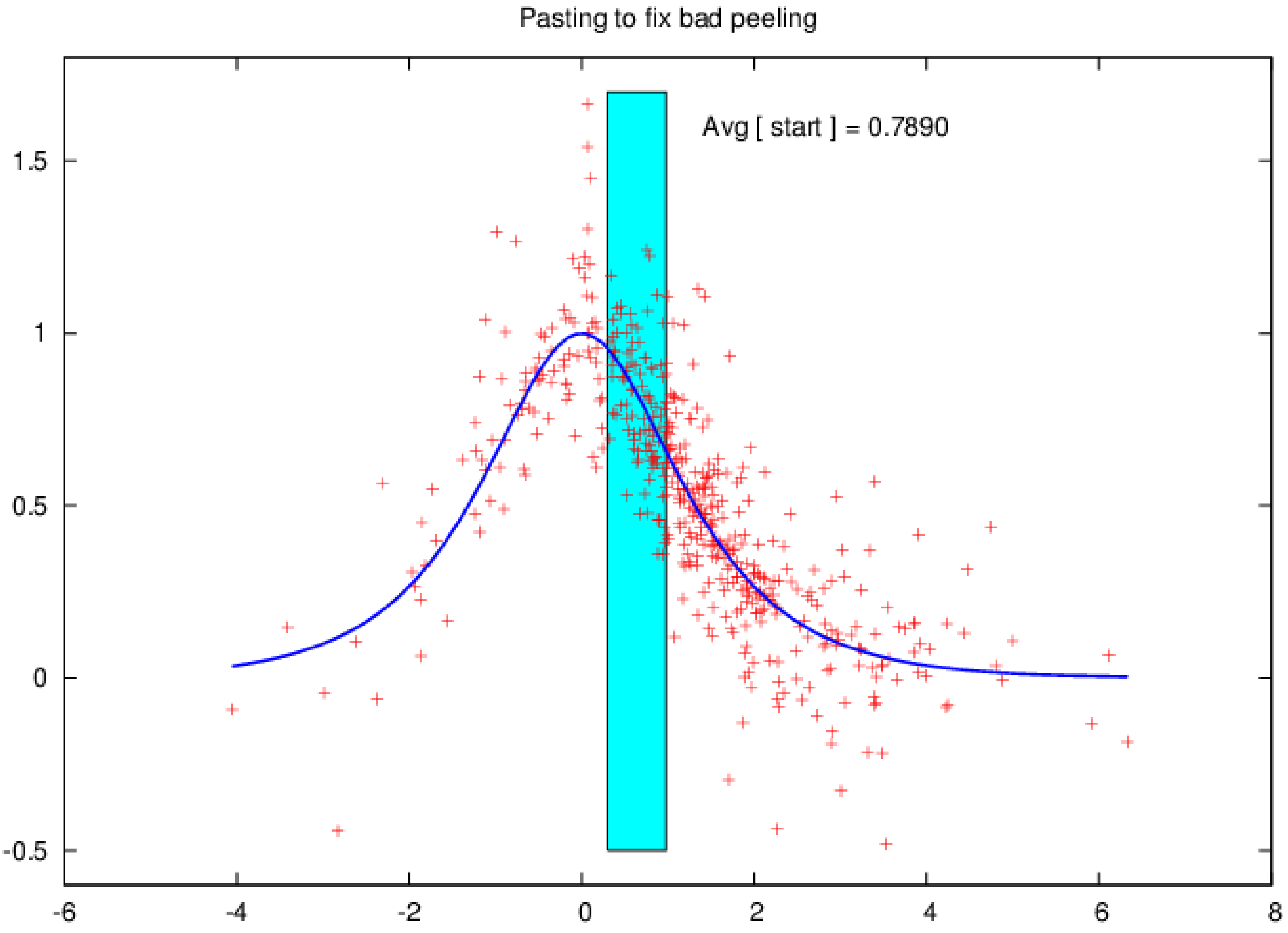
These images show pasting with α = 0.04 (20 points)

Note $<y_i>$



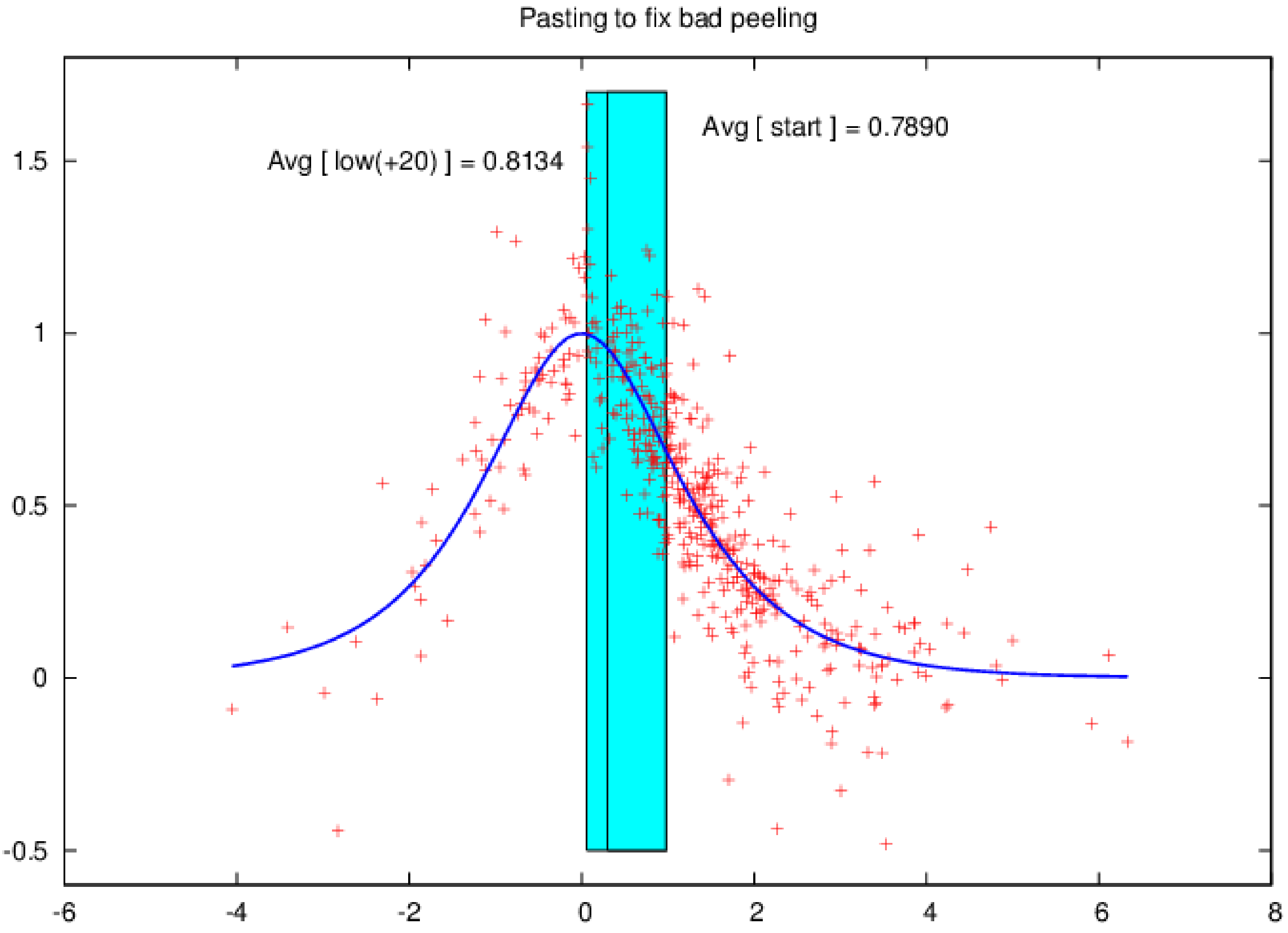
Example 2b

Step 1: **Calculate new <y> for adding α (0.04, 20 pts) to low and high ends**



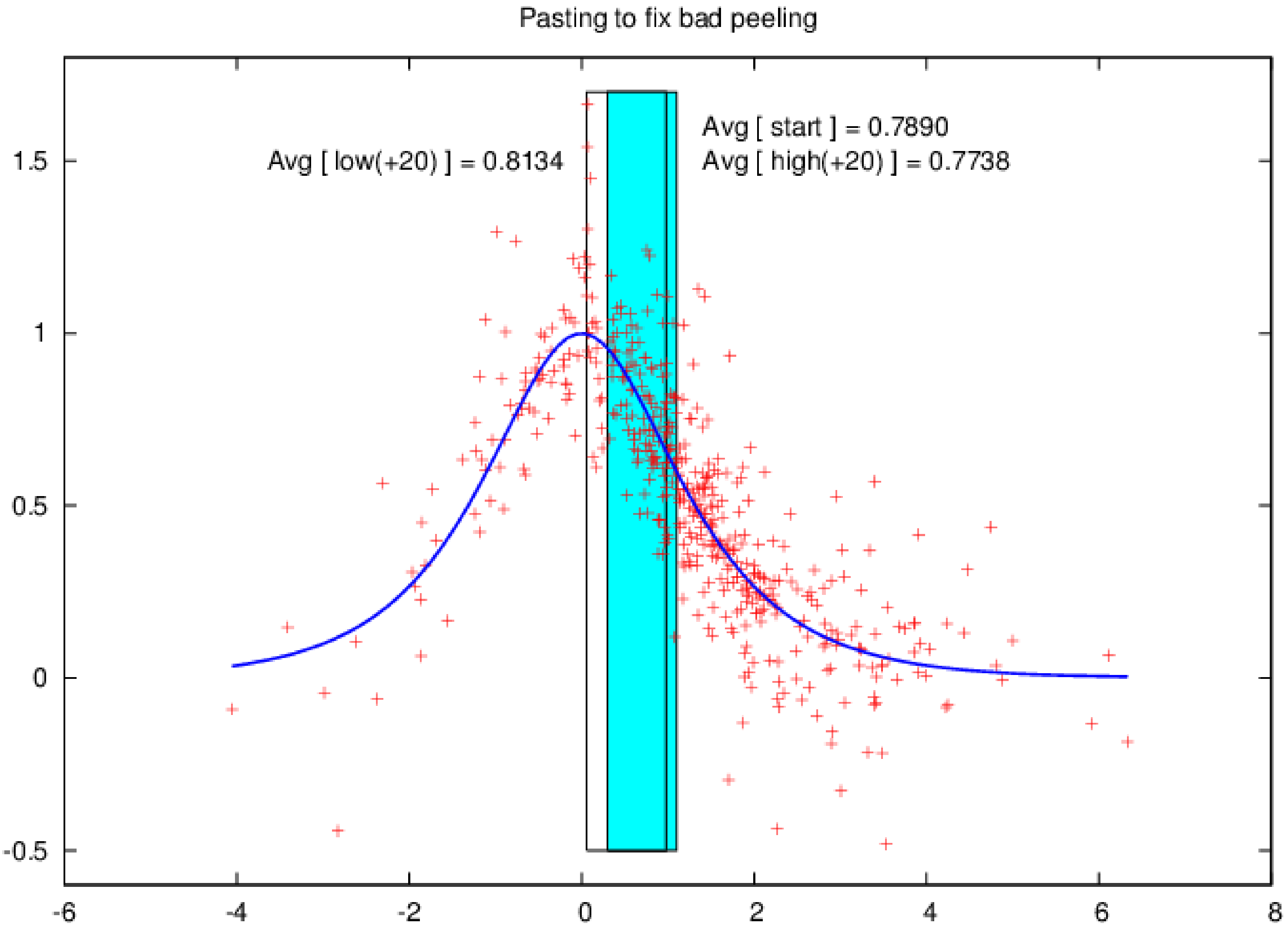
Example 2b

Step 1a: New <y> for adding α (0.04, 20 pts) **to low end**



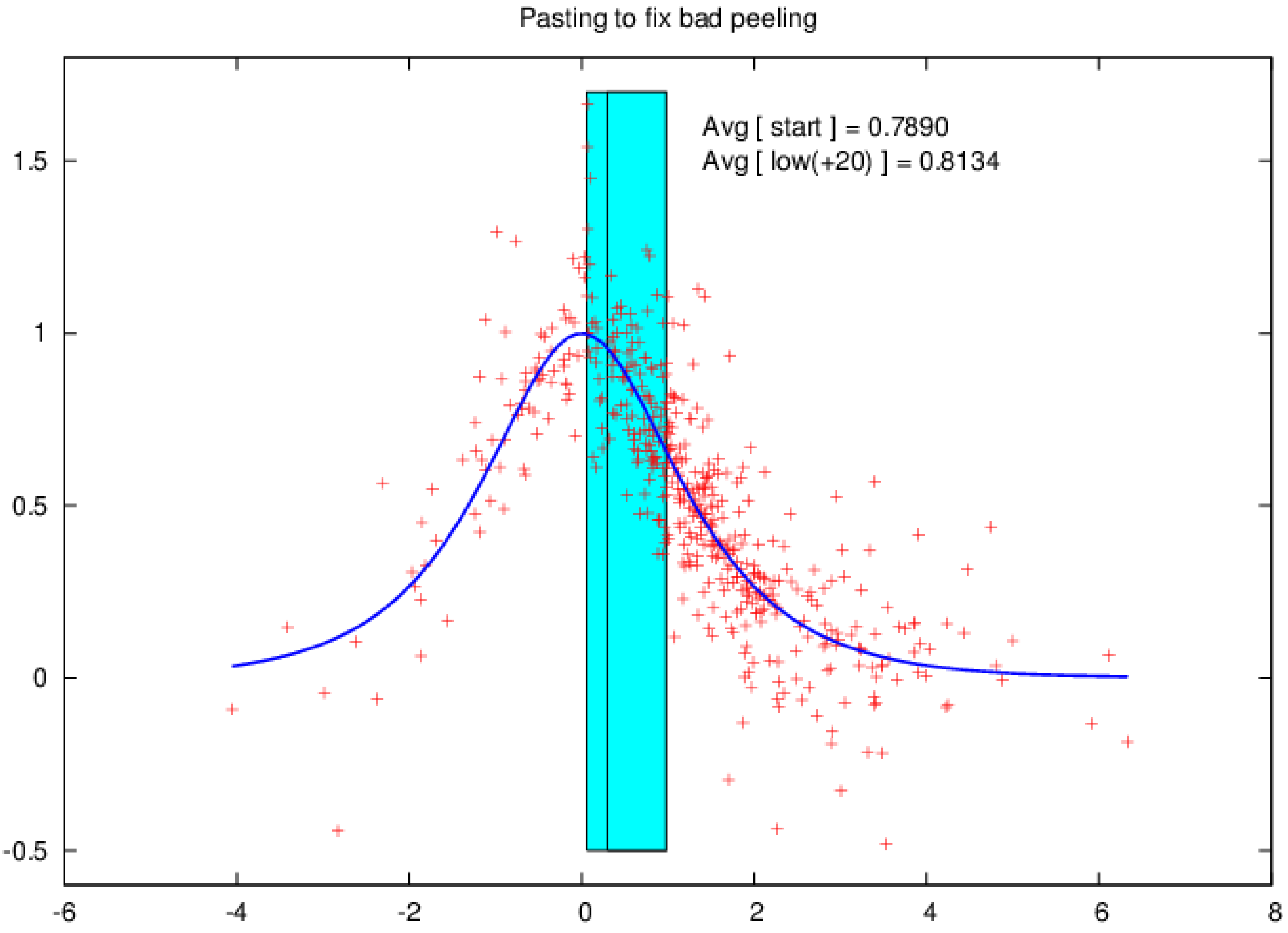
Example 2b

Step 1a: New <y> for adding α (0.04, 20 pts) **to high end** **(choose low!)**



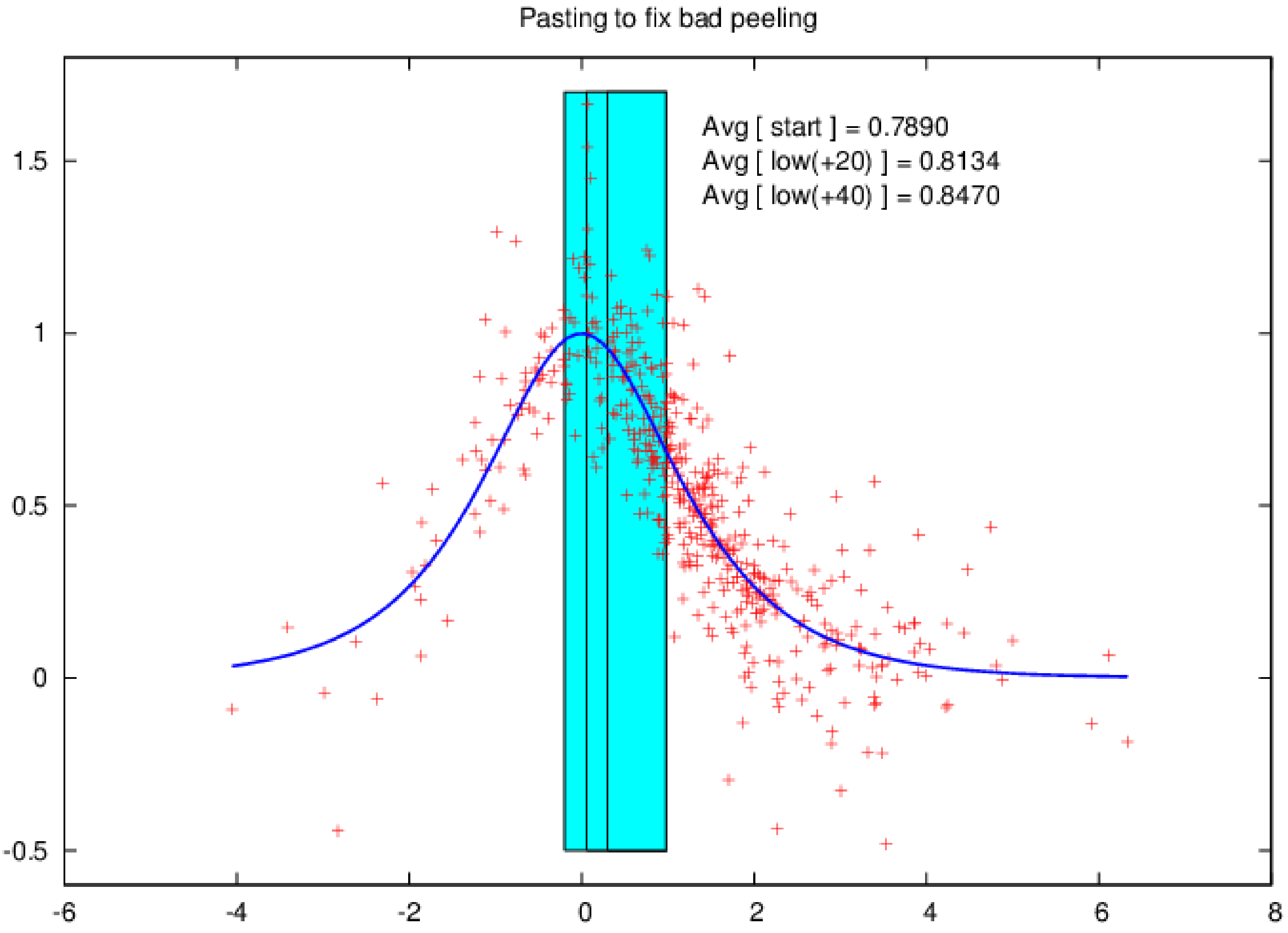
Example 2b

Step 2: **Repeat until <y> drops on both side**s (already dropped high).



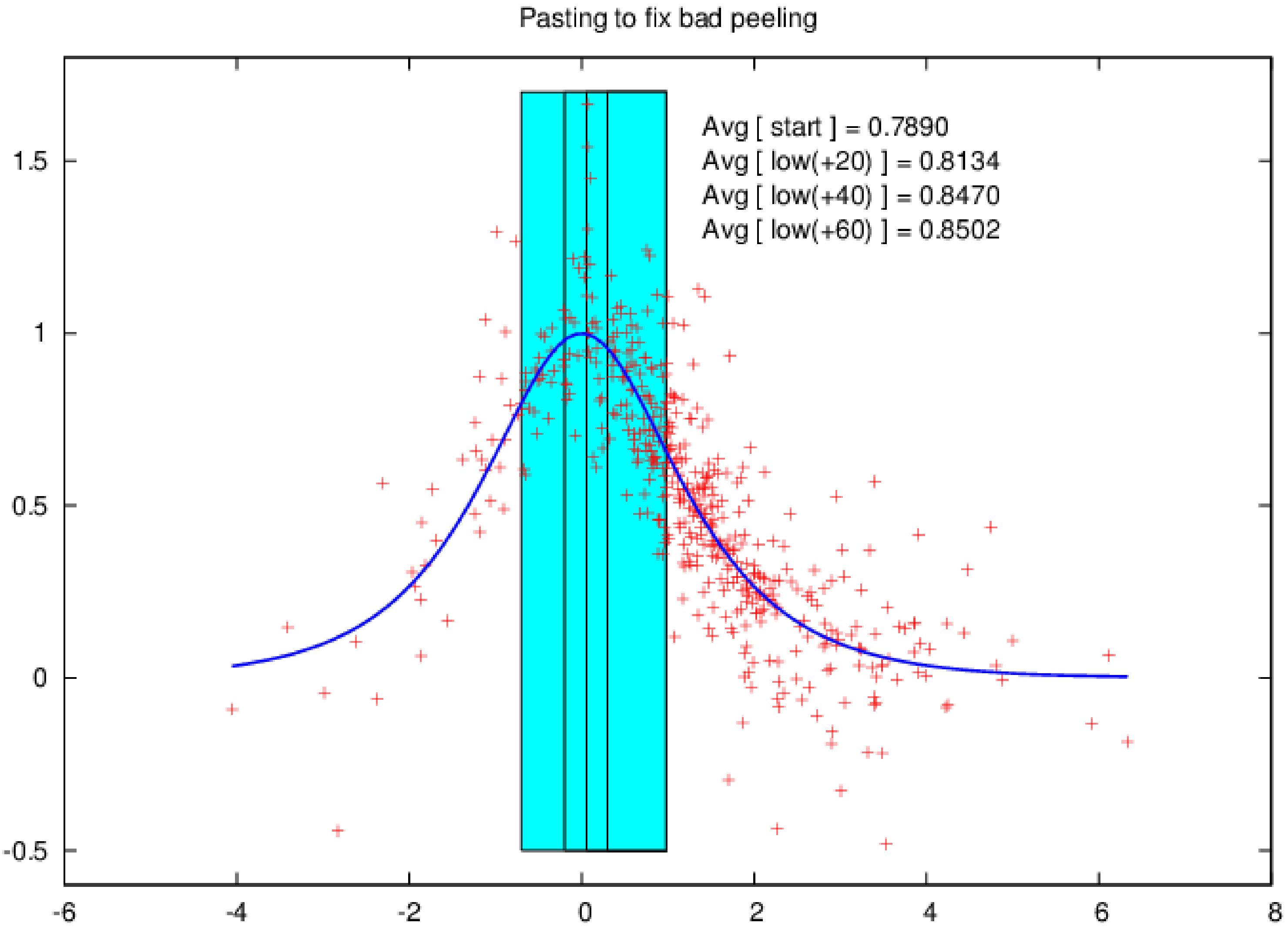
Example 2b

Step 2: Repeat until <y> drops on both sides (already dropped high).



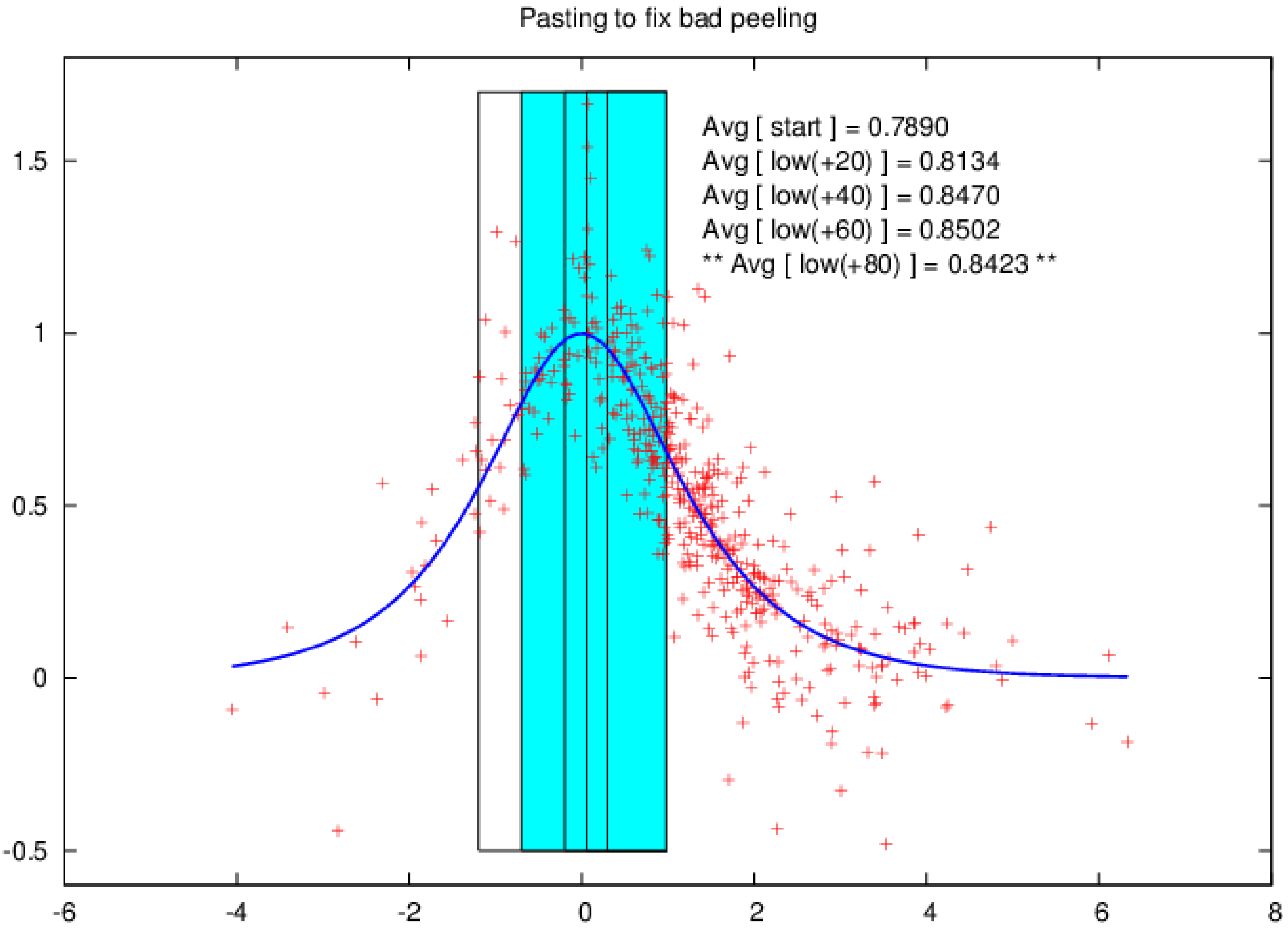
Example 2b

Step 2: Repeat until <y> drops on both sides (already dropped high).



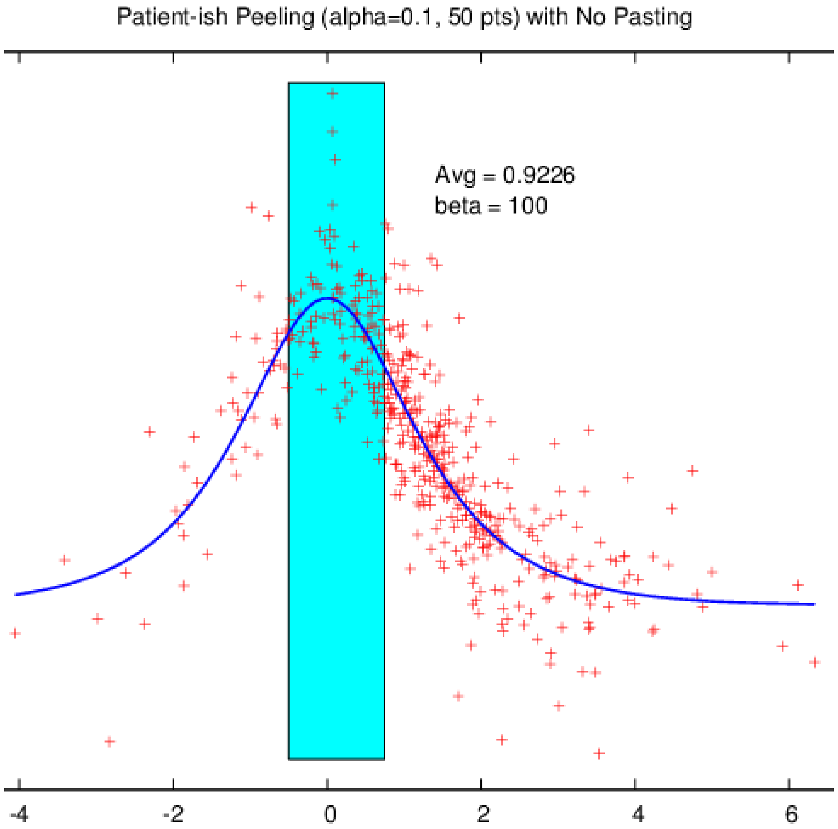
Example 2b

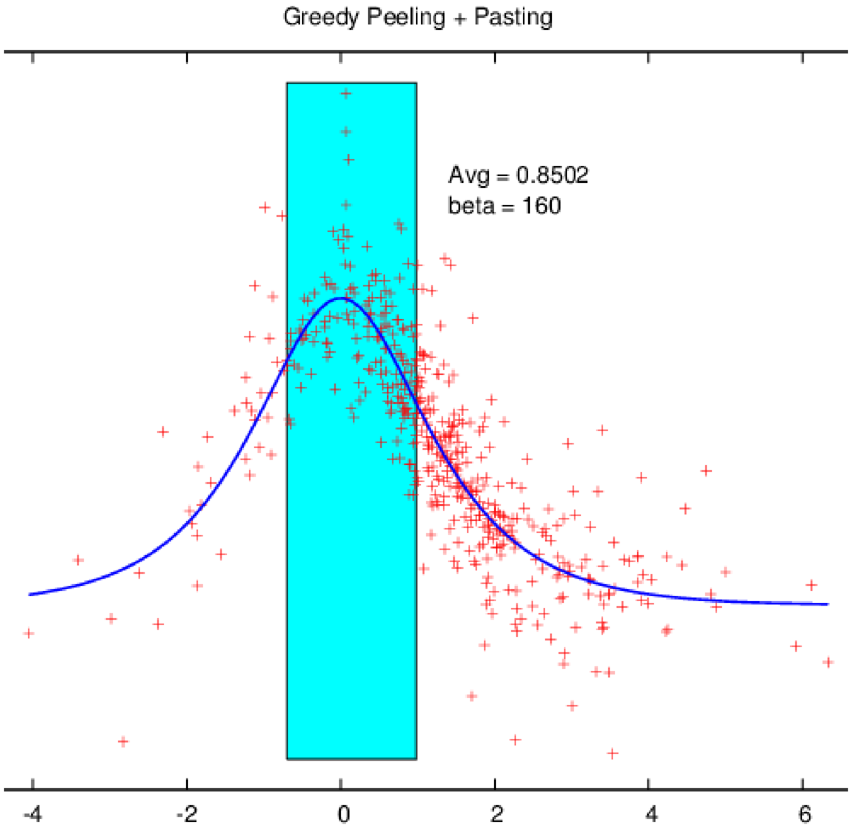
Step 3: **Stop adding when <y> drops on both sides** (already dropped high).



Compare results from Examples

Step 3: Stop adding when <y> drops on both sides (already dropped high).


Patient-ish Peeling (alpha=0.1, 50 pts) with No Pasting

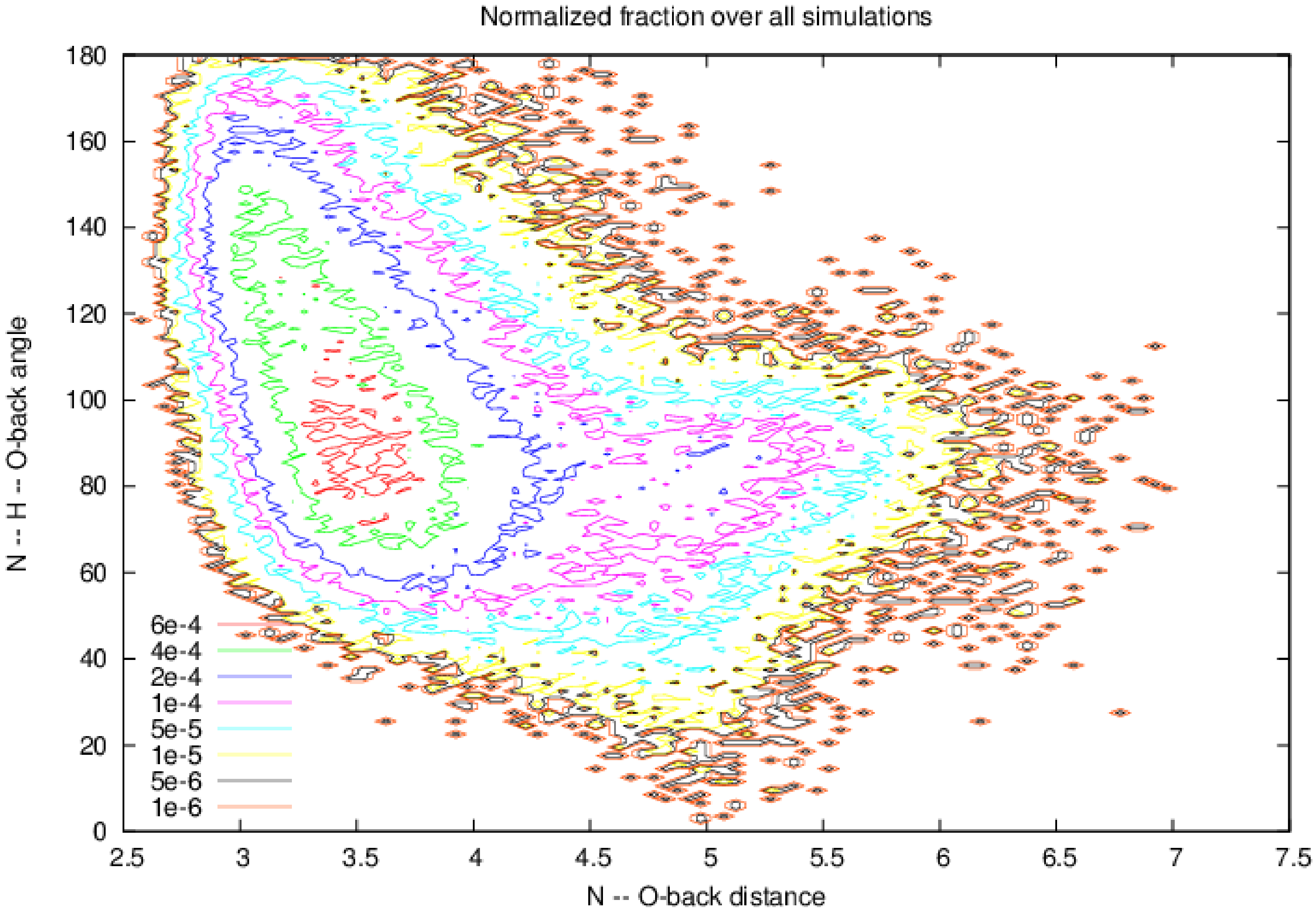

Greedy Peeling + Pasting

Many additional things in paper

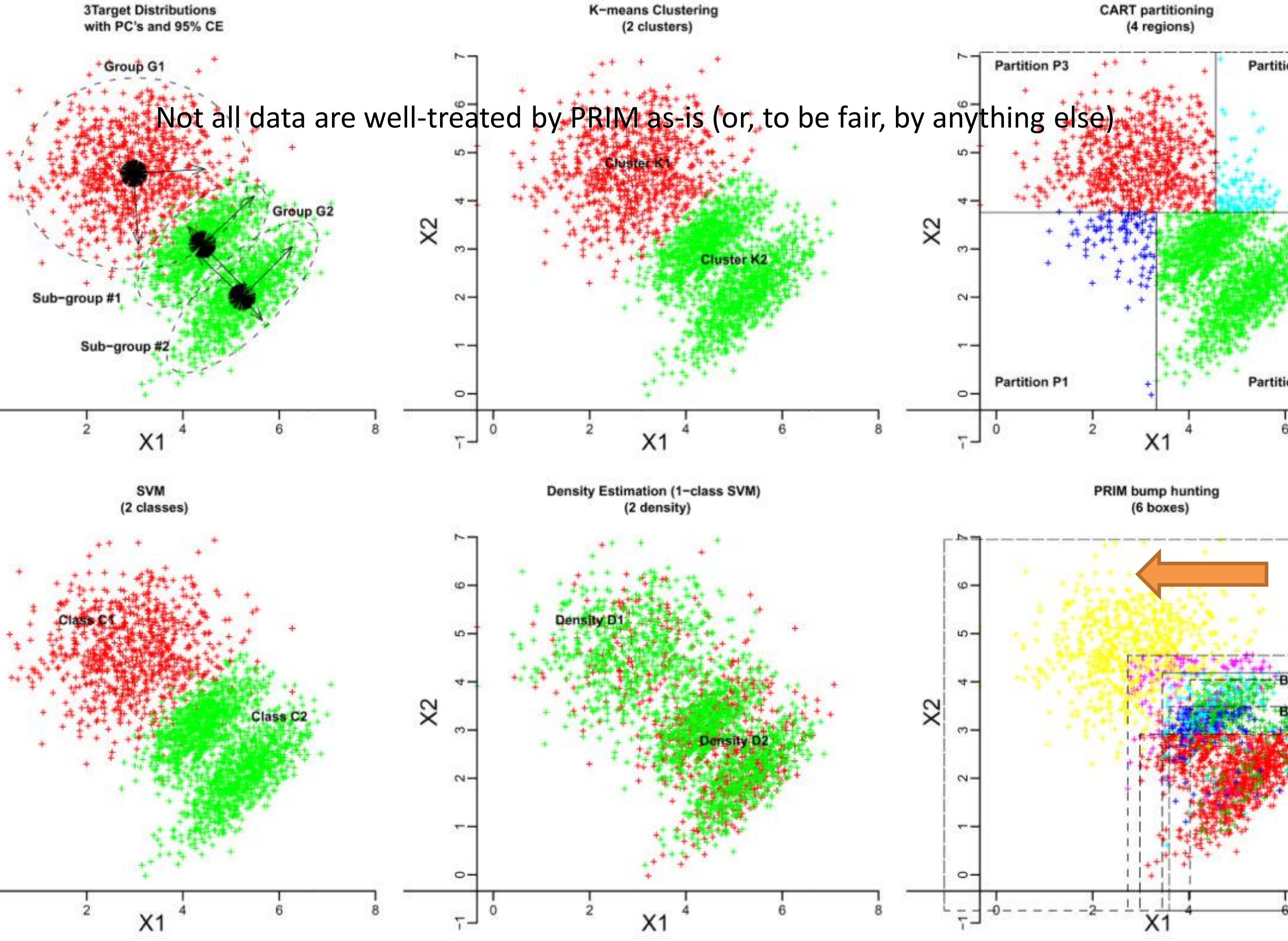
- Combine multiple runs
  - Vary α's and β's
- Of course, do this in multiple dimensions
  - Treat each axis separately
  - PRIM = Patient Rule Induction Method
- Vary protocols per user needs
  - For example, stop at a minimum range in x_i
    - β might still be used if reached first
    - Avoids solutions on domains that are impractically small
  - Also variants of, for example, peeling
- Several examples of different ways to use

Keep in mind our data are also large…

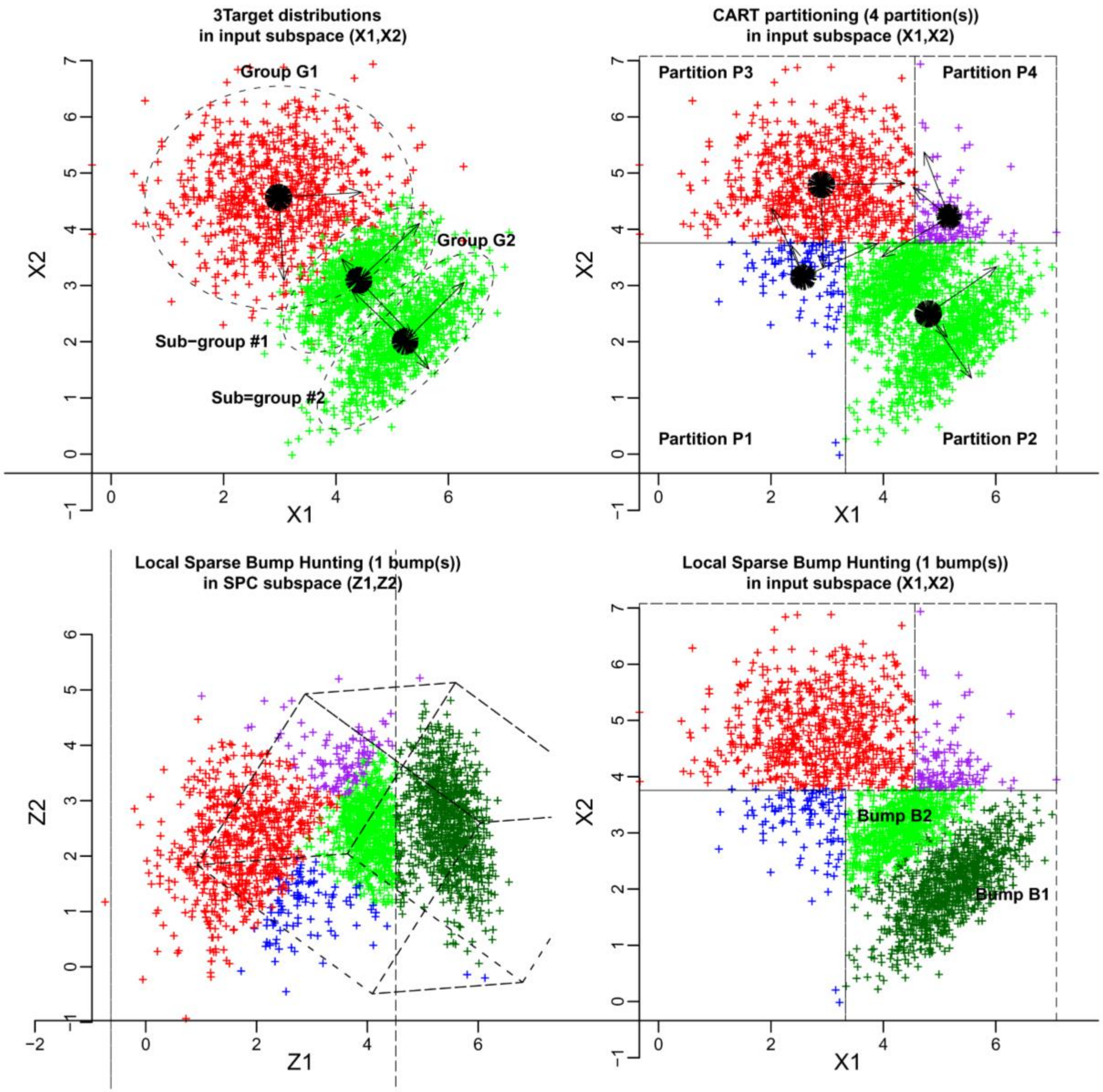
Regarding α, note tiny fractions of data in each region



Not all data are well-treated by PRIM as-is (or, to be fair, by anything else)



They use CART to get better coordinates



Questions?