Python で体験するベイズ推論
3.1

秋山研究室　学部4年　久保田陸人

2017/11/07

今日の内容

- 3. MCMCのなかをのぞいてみよう Opening the Black Box of MCMC
  - 3.1. 山あり谷あり、分布の地形
    - 3.1.1. MCMCで地形を探索する
    - 3.1.2. MCMCを実行するアルゴリズム
    - 3.1.3. 事後分布を近似する他の方法
    - 3.1.4. 例題：混合モデルの教師なしクラスタリング
    - 3.1.5. 事後サンプルを混ぜないで
    - 3.1.6. MAPを使って収束を改善

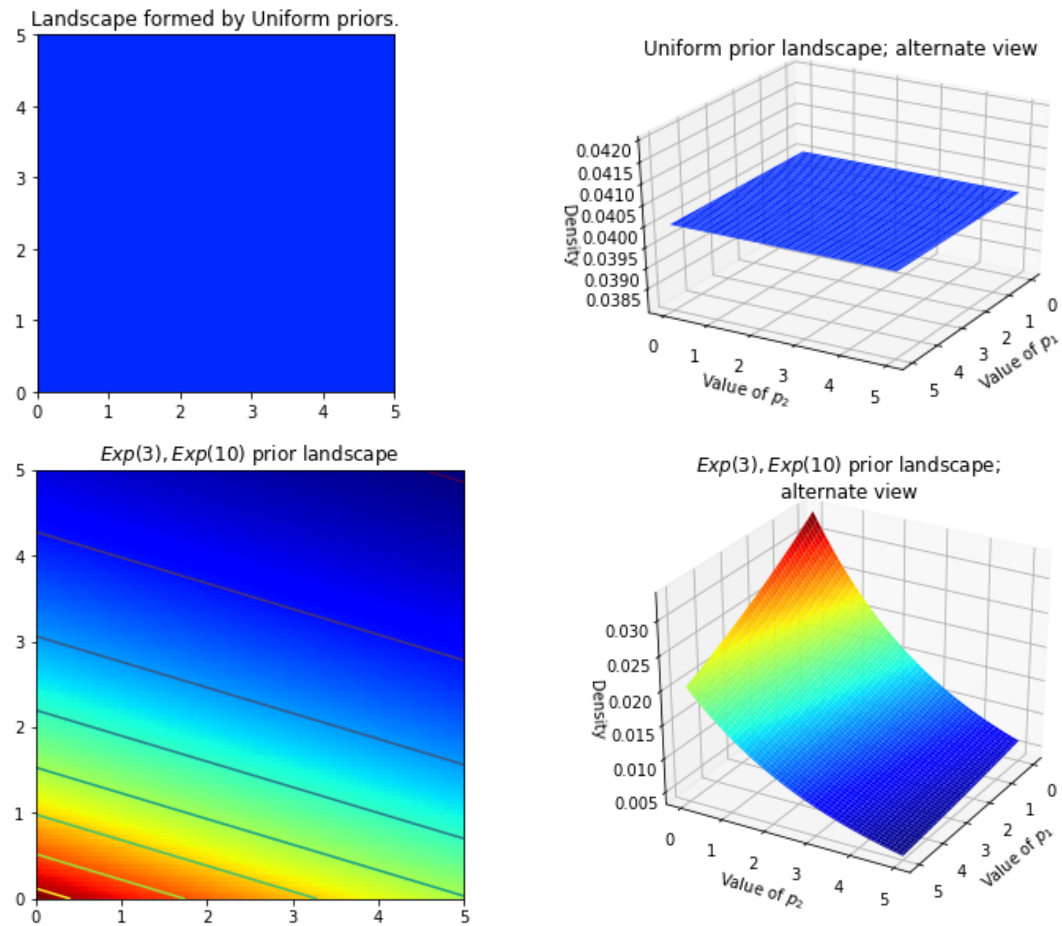
- （この章には演習問題は存在しませんでした）

分布の「地形」

- 未知数 p_1, p_2 に対して

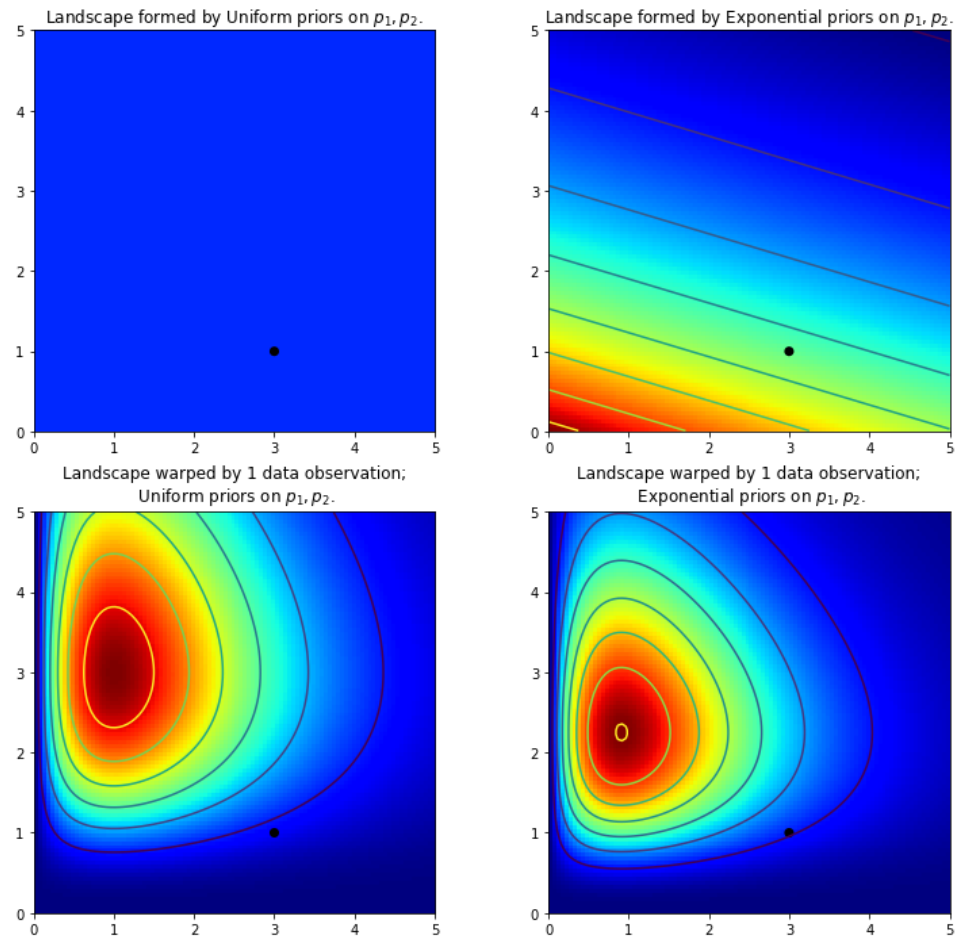
それらの事前分布を
[0, 5] の一様分布としたとき

それらの事前分布を
指数分布としたとき



分布の「地形」

- 事前分布（上）と、観測データがひとつ与えられたときの事後分布（下）の「地形」



  - 観測データは同じものだが、事前分布の影響で異なる事後分布になっている

MCMC とは

- マルコフ連鎖モンテカルロ法
  - モンテカルロ法
    - シミュレーションや数値計算を乱数を用いて行う手法の総称
      - よく円周率の近似値とか求めているやつ
  - マルコフ連鎖
    - 状態の遷移を確率的に行うもの
      - 前の状態から次の状態が決まる

  - モンテカルロ法は完全にランダムなので、計算コストがかさみ、精度も向上しない
  - マルコフ連鎖を定常分布としたサンプリングを行うことで改善

MCMC とは

- PyMC では、事前確率から事後確率を求めるのに MCMC を用いている

ベイズの定理

ベイズの定理：

$$P(A \mid X) = \frac{P(X \mid A)\,P(A)}{P(X)}$$

Where: データ: X
事象: A

$P(A)$: 事象Aが成立する確率
$P(X)$: データXが取れる確率
$P(A \mid X)$: データXが与えられた時事象Aが起こる確率
$P(X \mid A)$: 事象Aが成立する時データXが取れる確率

$P(X)$は事象Aによらないため

$$\underbrace{P(A \mid X)}_{\text{事後確率}} \propto P(X \mid A)\,\underbrace{P(A)}_{\text{事前確率}}$$

MCMC とは

- 乱数なんて使わなくても、計算して導出すればよくない？

  → N 次元空間中の曲面がどんなに起伏に富んでいようとも数式で表さなければならなくなり、これは簡単ではない

- 計算はできないけど、曲面の起伏が知りたい……

  → そこで**乱数**！！！

- 乱択でサンプリングを行うことにより、事後分布を推定できる

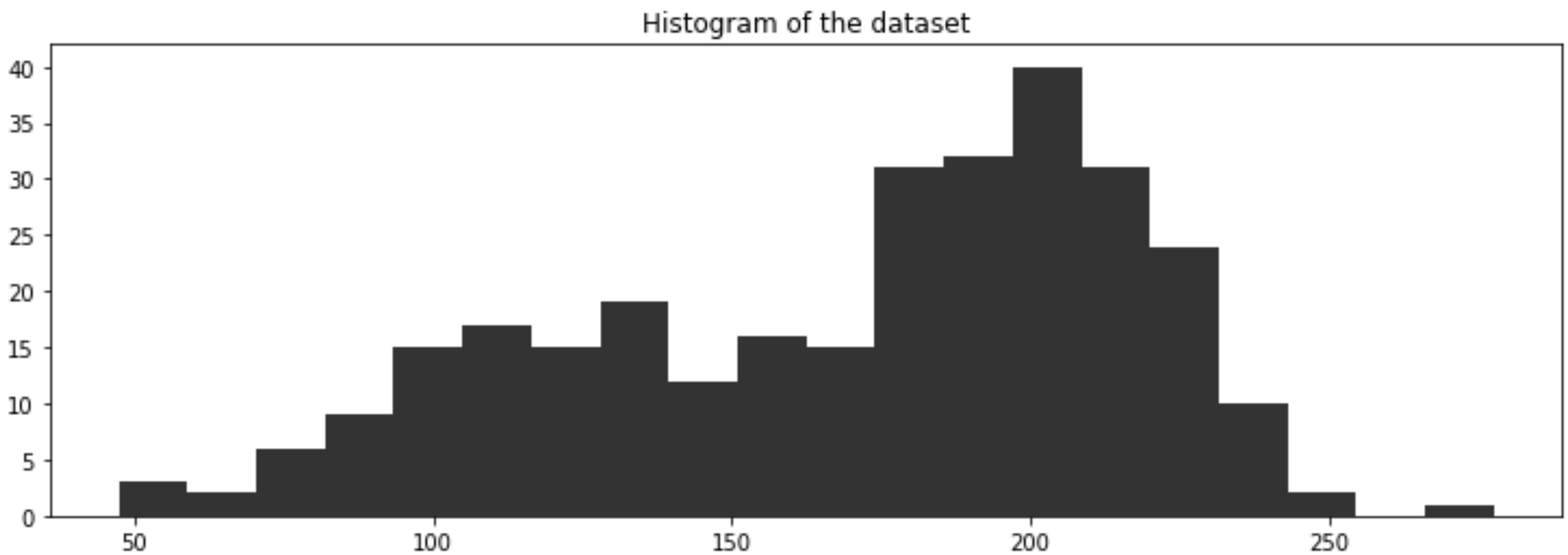
MCMC とは

- MCMC のアルゴリズムの大まかな概要
  - 1. 現在位置から始める
  - 2. 次に移動する先の位置を提案する
  - 3. その新しい位置がデータ及び事前分布に適しているか、を基準に新しい位置を受け入れる、または却下する
（**ここで乱数による選択が発生する**）
  - 4. (a) 受け入れたら：新しい位置へ移動し、1. へ戻る
(b) 却下したら　：今の位置にとどまり、1. へ戻る
  - 5. 以上を何度も繰り返し、全ての受け入れた位置を返す

  - 移動先の位置の提案方法、位置の取捨判断にはいくつかの手法があり、PyMC にもいくつか実装されている（らしい）

混合モデルの教師なしクラスタリング

- 以下のヒストグラムで表されるようなデータセットが与えられる
  - 山が 2 つある
  - 2 つのクラスタに分けられそう



混合モデルの教師なしクラスタリング

- 以下のようなデータ生成アルゴリズムをモデルとして考える
  - 1. 各データ点において、確率 p でクラスタ 0 を、そうでなければクラスタ 1 を選択し、i = (クラスタの番号) とする
  - 2. パラメータ μ_i と σ_i の正規分布から値を 1 つサンプリングする
  - 3. 繰り返す

- 事前分布のパラメータ
  - p： 0 から 1 の一様分布
  - μ_0 = 120, μ_1 = 190（データを眺めて得た推定値）
  - σ_i = 10

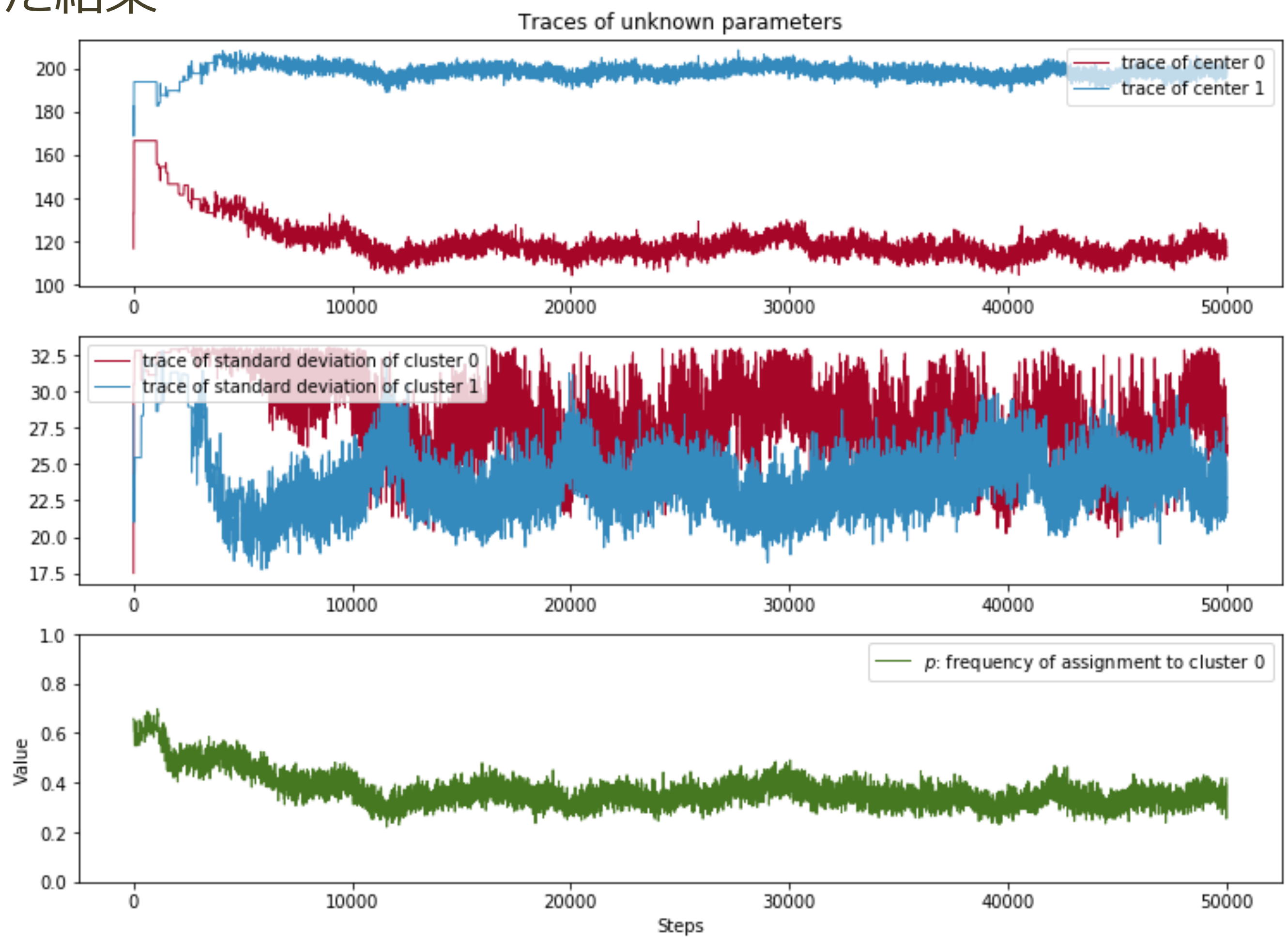
混合モデルの教師なしクラスタリング

- MCMC を回した結果

クラスタの中心

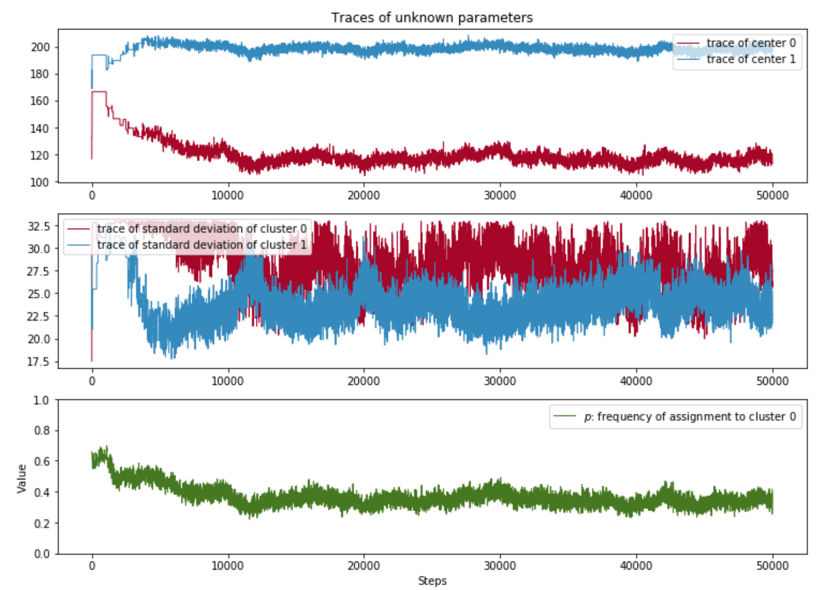
標準偏差

クラスタ 0 への
割当頻度 p



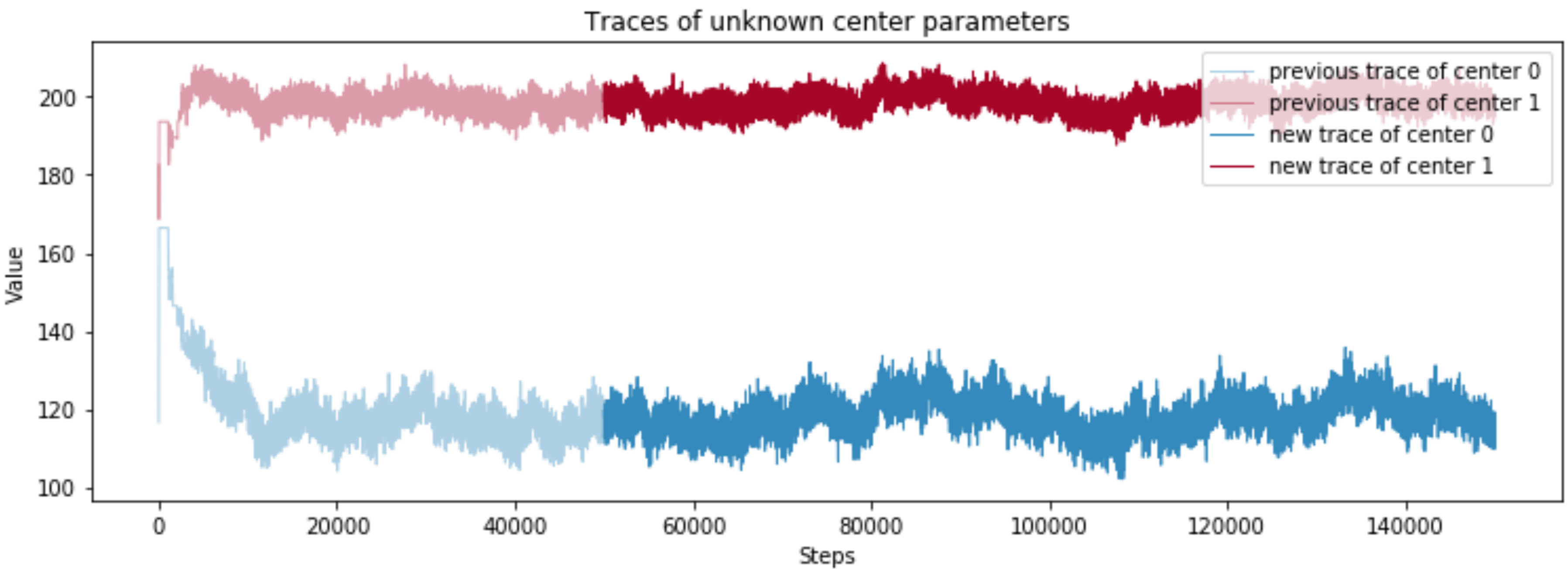
混合モデルの教師なしクラスタリング

- 以下のような性質が見て取れる
  - 軌跡は収束するが、ある一点に収束するのではなく、**ある分布に収束**する
  - 最初の数千点は、最終的にほしい分布とは関係がないため、それらは推論に使えないので捨てるべき
（この最初の数千点の期間を「バーンイン」と呼ぶ）



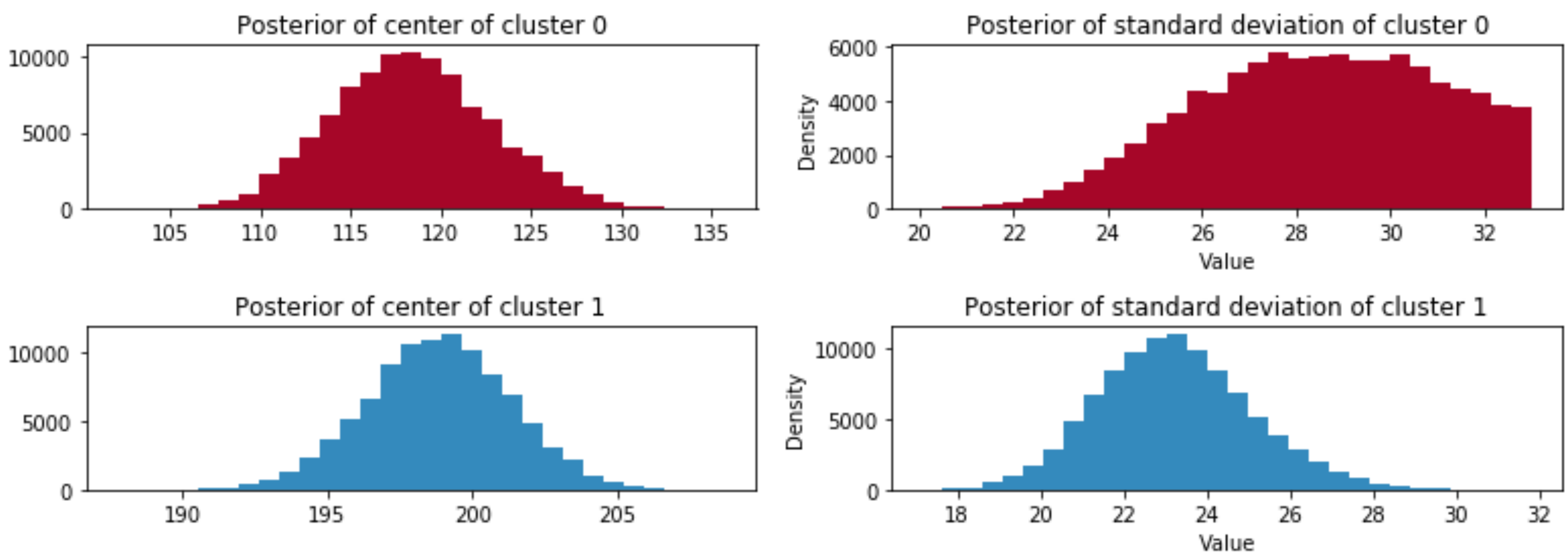
混合モデルの教師なしクラスタリング

- さらに 10 万回やった結果



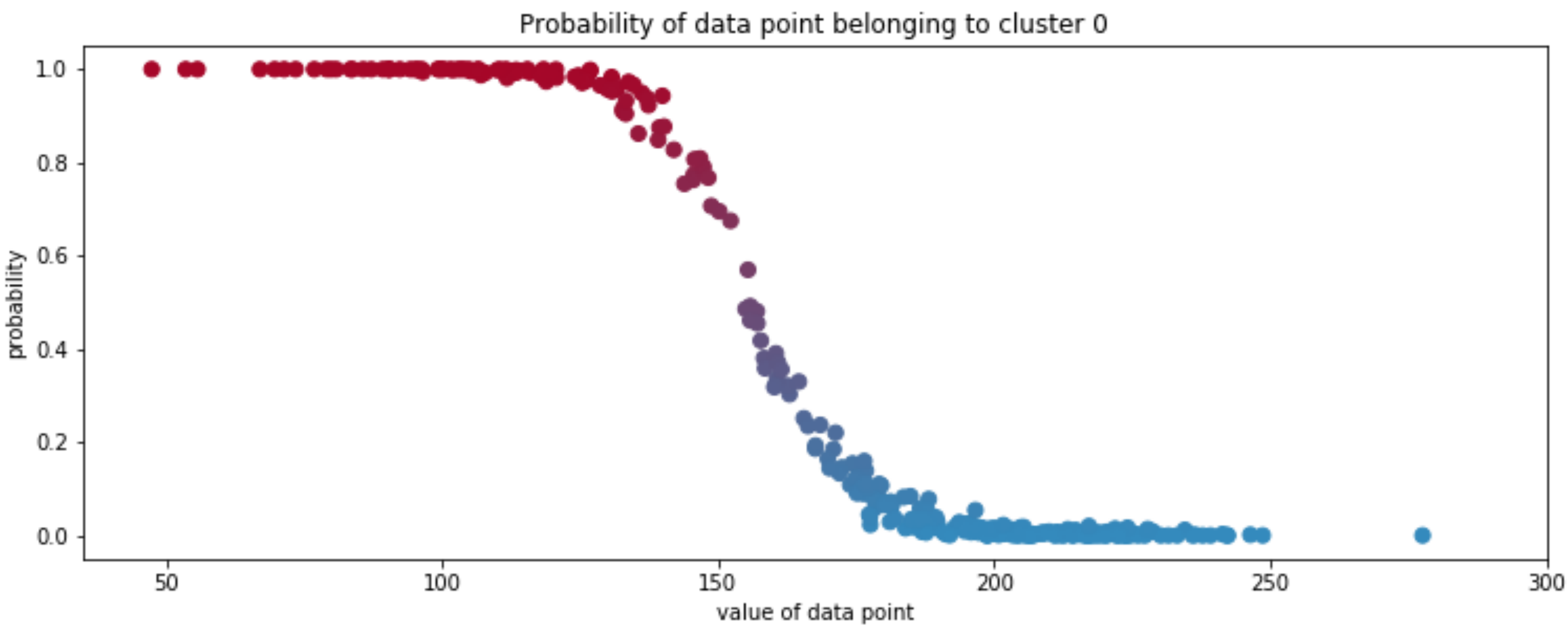
混合モデルの教師なしクラスタリング

- 各クラスタの中心と標準偏差の事後分布が得られたので、それをプロット



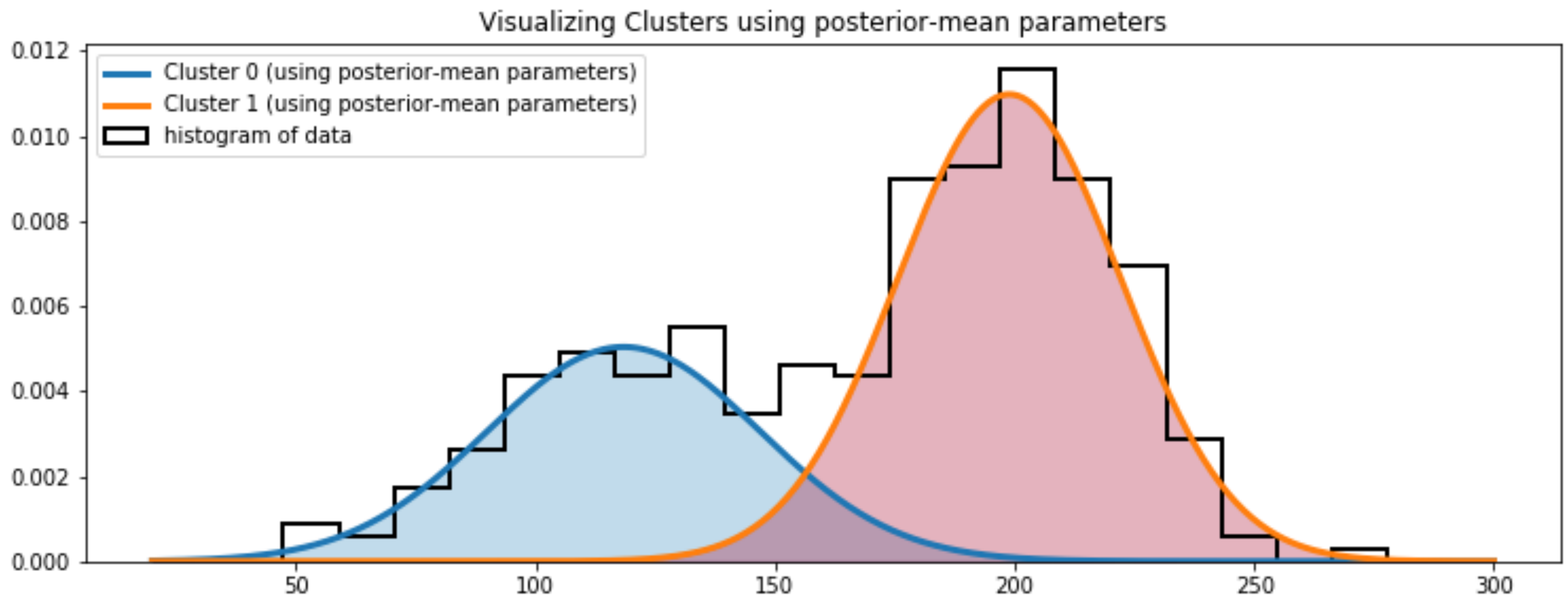
混合モデルの教師なしクラスタリング

- 事後分布より、各データ点がクラスタ 0 に属する頻度を推定してプロット



混合モデルの教師なしクラスタリング

- 得られたのは正規分布のパラメータの分布
  - できれば、データを最もよく表すある一つの正規分布がほしい
- 事後分布の平均を用いることで、簡単に良さげなものを拾うことができる



混合モデルの教師なしクラスタリング

- クラスタは得られたが、新たなデータ点が観測されたときに割り当てられるクラスタの予測はできるだろうか
  - x = 175 という点がクラスタ 1 に割り当てられる確率は、x に割り当てられるラベルを L_x とすると P(L_x = 1 | x = 175) となる
  - このデータを足してもう一度 MCMC を実行すれば推論は可能だが、データが 1 つ追加されるたびに大掛かりな処理が必要になる
  - しかし「x がクラスタ 1 に属する確率は、クラスタ 0 に属する確率より高いかどうか？」という問題はベイズの定理を用いて式変形を行うことにより比較的簡単に得られる
    - ここでは式は省略します

MAP について

- MAP (maximum a posterior) とは
  - 事後分布のピークのこと
  - 当然それを初期値としてそこからスタートした方が収束が早いし、バーンインの影響も抑えられる
  - PyMC には、MAP の位置を求める、または近似するメソッドがある