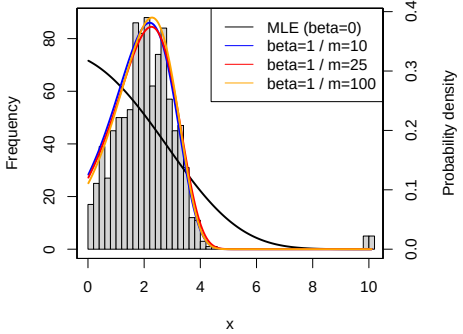
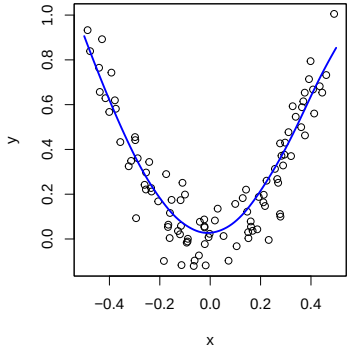
# [3GPM1-02] 積分型の損失関数を用いたパラメータ推定

奥野彰文[1,2]

[1]統計数理研究所, [2]理研AIP

# 何をしたのか




(A) 正規分布以外のロバスト推定

$$\min_{\theta}\left\{-\frac{1}{\beta}\frac{1}{n}\sum_{i=1}^{n}p_\theta(x_i)^\beta+\frac{1}{1+\beta}\int p_\theta(x)^{1+\beta}\mathrm{d}x\right\}$$

(B) 高次変動正則化

$$\min_{\theta}\left\{\frac{1}{n}\sum_{i=1}^{n}\{y_i-f_\theta(x_i)\}^2+\sum_{k=0}^{K}\eta_k\int\left|\frac{\partial^k f_\theta(x)}{\partial x^k}\right|^q\mathrm{d}x\right\}.$$

# (A) 正規分布以外のロバスト推定 (Okuno, AISM2024)

# 分布の推定

- 観測値$x_1, x_2, \ldots, x_n$が従う分布$Q$を知りたい.

経験分布$\hat{Q}_n(x) = \frac{1}{n}\sum_{i=1}^{n} \mathbb{1}(x_i \leq x)$と確率モデル$P_\theta$のKLダイバージェンス(=乖離度)

$$D(\hat{Q}_n, P_\theta) = -\frac{1}{n}\sum_{i=1}^{n} \log p_\theta(x_i) + \text{Const.}$$

を最小化して$P_\theta$が求まる. 最尤推定と等価.

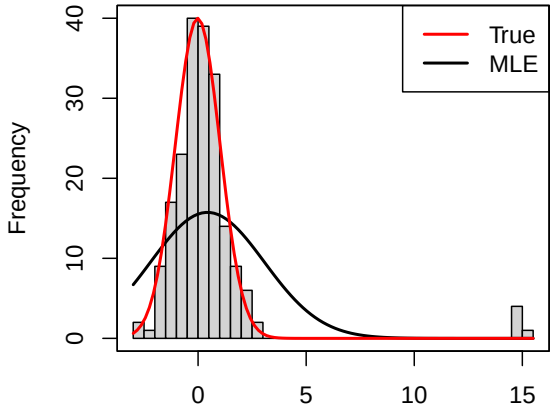



Figure: 最尤推定が外れ値に影響を受けるという典型的な図

# ロバストな分布の推定

冪密度ダイバージェンス (Density power divergence, DPD; Basu et al. 1998):

$$D_\beta(\hat{Q}, P_\theta) = -\frac{1}{\beta}\frac{1}{n}\sum_{i=1}^{n} p_\theta(x_i)^\beta + \frac{1}{1+\beta}\int p_\theta(x)^{1+\beta}\mathrm{d}x + \text{Const.}$$

を最小化するとよい.

特に $\beta \searrow 0$ のときKLダイバージェンス($\Leftrightarrow$ 最尤推定)になる.

正規分布の場合，$\xi = 0.1$.

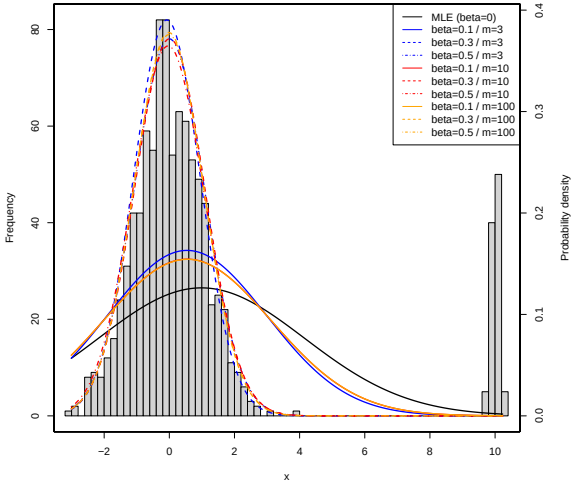



Figure: 外れ値の影響を軽減できた

# 研究を始める前の Q & A

Q. 外れ値を目視で外せばよいのでは

A. 私もそう思います．希望があるとすれば高次元*や時系列，自動化の文脈など...

Q. 本当に「ロバスト」な推定なんですか

A. ロバストって結局何なんでしょうか ⇒ https://arxiv.org/abs/2407.10418

Q. Huberみたいな昔ながらのロバスト推定ではダメですか

A. 平均等だけでなく，分布そのものを推定できることが強みです．

Q. 計算コストはどうですか

A. 正規分布など特定の分布以外での最適化は厳しいです．

# 最適化のボトルネック

$$D_\beta(\hat{Q}, P_\theta) = -\frac{1}{\beta}\frac{1}{n}\sum_{i=1}^{n} p_\theta(x_i)^\beta + \underbrace{\frac{1}{1+\beta}\int p_\theta(x)^{1+\beta}\mathrm{d}x}_{=:r_\theta^{(\beta)}} + \mathrm{Const.}$$

- $P_\theta = N(\mu, \sigma^2)$とすると，$r_\theta^{(\beta)} = (2\pi\sigma^2)^{-\beta/2}(1+\beta)^{-3/2}$.
- 計算できる：正規，一般化指数，一般化パレート，ワイブルモデル.
- それ以外で計算できない．例えば一般の指数型分布モデルで計算できない
  - 混合正規モデルでの近似最適化 (Fujisawa and Eguchi, 2006),
  - ポアソンモデルでの近似最適化 (Kawashima and Fujisawa, 2019),
  - 歪正規モデルでの近似最適化 (Nandy et al., 2022).

分布推定できることが強みなのに，最適化できる確率モデルが少ない矛盾

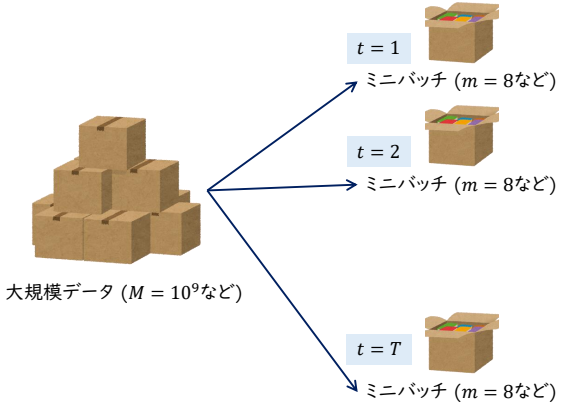
# 数値積分ではダメなのか

- 勾配法+数値積分: $y_1^{(t)}, y_2^{(t)}, \cdots \sim p_{\theta^{(t)}}$ について

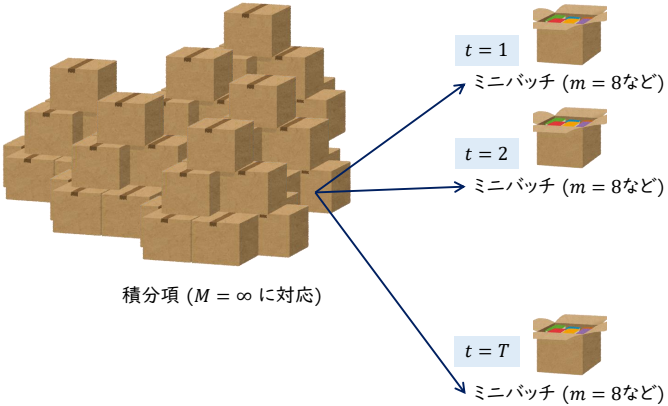
$$\begin{aligned}\theta^{(t+1)} &\leftarrow \theta^{(t)} - \gamma \nabla D_\beta(\hat{Q}, P_{\theta^{(t)}}) \\ &= \theta^{(t)} - \gamma \left\{ -\frac{1}{\beta}\frac{1}{n}\sum_{i=1}^{n} \nabla\{p_{\theta^{(t)}}(x_i)^\beta\} + \frac{1}{1+\beta} \operatorname*{plim}_{M\to\infty} \frac{1}{M}\sum_{j=1}^{M} \nabla\{p_{\theta^{(t)}}(y_j^{(t)})^\beta\} \right\}\end{aligned}$$

- 各反復で数値積分が必要：次元$d$が増えると必要な$M$は大きくなる $(M \sim \varepsilon^{1/d})$
- 学習率 $\gamma$ は固定，数値積分のサンプルサイズ $M$ は可能な限り大きくする.
- 計算量$O(TM)$, 近似誤差は$T \to \infty$でも$O_p(1/\sqrt{M}) > 0$.

# ところで，深層学習で使われるミニバッチ最適化 ($M$ =巨大)



# 積分 $\Leftrightarrow M = \infty$



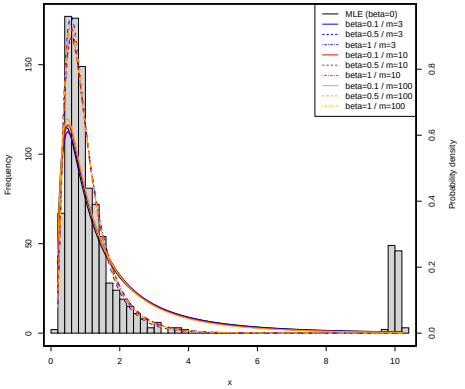
# 提案法

$y_1^{(t)}, y_2^{(t)}, \cdots \sim P_{\theta^{(t)}}$について

$$\theta^{(t+1)} \leftarrow \theta^{(t)} - \gamma^{(t)} \left\{ -\frac{1}{\beta}\frac{1}{n}\sum_{i=1}^{n} \nabla\{p_{\theta^{(t)}}(x_i)^\beta\} + \frac{1}{1+\beta}\frac{1}{m}\sum_{j=1}^{m} \nabla\{p_{\theta^{(t)}}(y_j^{(t)})^\beta\} \right\}$$

- 学習率を下げるのがポイント$\gamma^{(t)} \to 0$ $(t \to \infty)$.
- $m$は何でもよい．例えば$m = 8$. $m = 1$でもよい!
- 近似誤差 $O_p(1/\sqrt{M}) > 0$ のある数値積分と違い，真値に収束する! $\hat{\theta}^{(t)} \to \theta_*$.
- 最適化分野ではよく知られている: 確率的最適化 (Robbins and Monro, 1951).
- Contrastive divergence (Hinton et al., 2002).

逆ガウス分布の場合，$\xi = 0.1$.




$$p_\theta(x) = \sqrt{\frac{\lambda}{2\pi x^3}} \exp\left(-\frac{\lambda(x-\mu)^2}{2\mu^2 x}\right)$$

(*逆ガウス分布は積分項が明示的に計算できない)

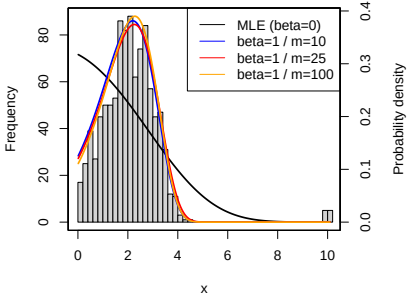



Figure: Gompertz分布のフィッティング

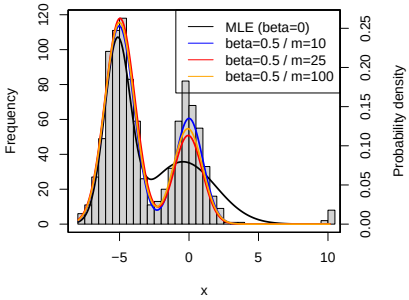



Figure: 混合ガウス分布のフィッティング

$$\text{Gompertz: } p_\theta(x) = \lambda \exp\left(\omega x + \frac{\lambda}{\omega}\{1 - \exp(\omega x)\}\right), \ (x \geq 0).$$

(*Gompertz分布も混合ガウス分布も積分項が明示的に計算できない)

# 追加の話

- パッケージ化 `https://github.com/oknakfm/sgdpd`


Figure: 10ページくらいの例題付きマニュアルも用意しています

- $\gamma$-divergence (Fujisawa and Eguchi, 2008) の最小化もできる.
- 最適化分野では当たり前の技術，だが統計ではあまり馴染みがない
- 他にも使えるのでは? $\Rightarrow$ (B) 高次変動正則化

## (B) 高次変動正則化 (Okuno, arXiv:2308.02293)

# 線形回帰におけるパラメータ正則化

線形回帰モデル$f_\theta(x) = \langle \theta, x \rangle$のパラメータ正則化とは，

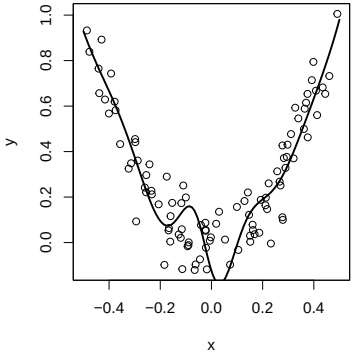
$$\|\theta\|_2^2 = \left\| \frac{\partial f_\theta(x)}{\partial x} \right\|_2^2, \quad \forall x.$$

つまり入力$x$に対する応答$y = f_\theta(x)$の感度を抑えている．

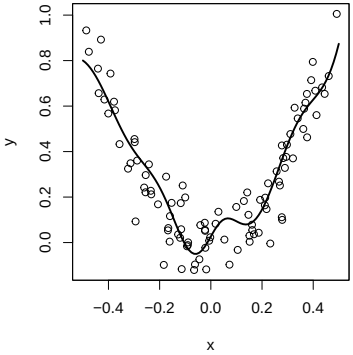
- $f_\theta$が非線形なら，パラメータ正則化する意味は?
- パラメータの特異性を局所的に解決したりはするが，変動を抑えると限らない．

# 特異性は抑えるが変動は抑制されない

$f_\theta(x)$は単層パーセプトロンとする.



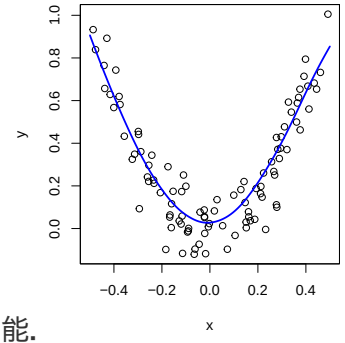

(a) 正則化なし




(b) リッジ正則化

# 提案法: 高次変動正則化

$$\frac{1}{n}\sum_{i=1}^{n}\{y_i - f_\theta(x_i)\}^2 + \sum_{k=0}^{K}\eta_k \int \left|\frac{\partial^k f_\theta(x)}{\partial x^k}\right|^q \mathrm{d}x.$$




- (A)と同様にして最適化可能.
- Physics-informed training of NN (PINN) の別実装とも思える.

# Q & A

Q. スプラインでよいのでは?

A. 今の主流はニューラルネットです．簡単に実装できることが大事.

A. 判別境界の正則化など，何にでも使いまわせて汎用性が高い.

Q. ニューラルネットの汎化の話との関係は?

A. 元のサンプルサイズが大きければバッチ化などにより勝手に正則化の効果が生じる，というのが最近の主流の解釈かと思います．統計の問題のように，小サンプルサイズの場合にのみ必要な技術です.

Q. 計算が重いのでは?

A. 自動微分ベースだとPINNと同じくらいには重いです．単純なNNなら微分を明示的に書いてしまえば高速に学習可能です.

# まとめ

(1) Akifumi Okuno. Minimizing robust density power-based divergences for general parametric density models. AISM. 2024.
(一般の確率モデルでのロバストダイバージェンス最小化)

(2) Akifumi Okuno. A stochastic optimization approach to train non-linear neural networks with a higher-order variation regularization. arXiv:2308.02293
(一般の非線形モデルでの高次変動正則化)

質問やコメントは okuno@ism.ac.jp まで.