

# ハイパーパラメータ推定

岡山大学 異分野基礎科学研究所

大槻純也



# ハイパーパラメータ

## ベイズ線形回帰

予測分布

$$p(t|\mathbf{x}, \mathbf{w}) = \mathcal{N}(t|y(\mathbf{x}, \mathbf{w}), \beta^{-1})$$

測定誤差

事前分布 (正則化)

$$p(\mathbf{w}) = \mathcal{N}(\mathbf{w}|\mathbf{0}, \alpha^{-1})$$

正則化パラメータ

基底関数

$$\phi(\mathbf{x}|\boldsymbol{\theta})$$

基底パラメータ

---

## ガウス過程

カーネル関数

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_1 \exp\left(-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{2\theta_2}\right) + \beta^{-1}\delta_{nm}$$

測定誤差

---

## LASSO

$$\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\}$$

正則化パラメータ

# ハイパーパラメータ推定

- 解析的方法

エビデンス近似または第二種最尤推定と呼ぶ。

具体的な評価式は、推定法（ベイズ線形回帰、ガウス過程）ごとに異なる。

- エルボー法

二乗誤差のハイパーパラメータ依存性から決定するお手軽法。妥当なハイパーパラメータの値の目安を得る方法として有用。

- 交差検証法 (Cross Validation)

どの推定法（ベイズ線形回帰、ガウス過程、LASSO）にも使える汎用的な評価法。

ただし、計算量が多い。並列計算可能ではある。

# エビデンス近似 (線形回帰の場合)

これまではハイパーパラメータは固定していた

$$p(t|\mathbf{t}, \alpha, \beta) = \int p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) d\mathbf{w}$$

条件                      重みに関して周辺化

完全にベイズ統計の手続きに従うなら、重み  $\mathbf{w}$  と同様に、ハイパーパラメータに関する積分 (周辺化) するのが最善

$$p(t|\mathbf{t}) = \iiint p(t|\mathbf{w}, \beta) p(\mathbf{w}|\mathbf{t}, \alpha, \beta) p(\alpha, \beta|\mathbf{t}) d\mathbf{w} d\alpha d\beta$$

ハイパーパラメータの事後分布

ベイズの定理

$$p(\alpha, \beta|\mathbf{t}) \propto p(\mathbf{t}|\alpha, \beta) p(\alpha, \beta)$$

尤度関数   事前分布

しかし、解析的に積分を実行することは難しいので、特定の値に固定してしまうのが普通

$$p(t|\mathbf{t}) \simeq p(t|\mathbf{t}, \hat{\alpha}, \hat{\beta}) = \int p(t|\mathbf{w}, \hat{\beta}) p(\mathbf{w}|\mathbf{t}, \hat{\alpha}, \hat{\beta}) d\mathbf{w}$$

特定の値に固定

事前分布はフラットと仮定し、尤度関数を最大化する

$$p(\mathbf{t}|\alpha, \beta) \stackrel{\text{sum rule}}{=} \int p(\mathbf{t}|\mathbf{w}, \beta) p(\mathbf{w}|\alpha) d\mathbf{w}$$

周辺化尤度 ( $\mathbf{w}$  について周辺化してある)  
あるいはエビデンス (evidence)

$$\frac{\partial}{\partial \alpha} p(\mathbf{t}|\alpha, \beta) = 0, \quad \frac{\partial}{\partial \beta} p(\mathbf{t}|\alpha, \beta) = 0$$

第二種最尤推定あるいはエビデンス近似

# ガウス過程の場合

尤度関数

$$p(\mathbf{t}_N|\boldsymbol{\theta}) = \mathcal{N}(\mathbf{t}_N|0, C_N)$$

$$= \frac{1}{(2\pi)^{N/2}} \frac{1}{|C_N|^{1/2}} \exp\left(-\frac{1}{2}\mathbf{t}_N^T C_N^{-1} \mathbf{t}_N\right)$$

カーネル関数

$$k(\mathbf{x}_n, \mathbf{x}_m) = \theta_1 \exp\left(-\frac{\|\mathbf{x}_n - \mathbf{x}_m\|^2}{2\theta_2}\right) + \beta^{-1} \delta_{nm}$$

対数尤度

$$\ln p(\mathbf{t}_N|\boldsymbol{\theta}) = -\frac{1}{2} \ln |C_N| - \frac{1}{2} \mathbf{t}_N^T C_N^{-1} \mathbf{t}_N - \frac{N}{2} \ln(2\pi)$$

これを最大化するように  $\boldsymbol{\theta}$  を選ぶ

微分を計算

$$0 = \frac{\partial}{\partial \theta_i} \ln p(\mathbf{t}_N|\boldsymbol{\theta}) = -\frac{1}{2} \frac{\partial}{\partial \theta_i} \ln |C_N| - \frac{1}{2} \mathbf{t}_N^T \frac{\partial C_N^{-1}}{\partial \theta_i} \mathbf{t}_N$$

$$= -\frac{1}{2} \text{Tr}\left(C_N^{-1} \frac{\partial C_N}{\partial \theta_i}\right) + \frac{1}{2} \mathbf{t}_N^T C_N^{-1} \frac{\partial C_N}{\partial \theta_i} C_N^{-1} \mathbf{t}_N$$

公式

$$\frac{\partial}{\partial x} \ln |A| = \text{Tr}\left(A^{-1} \frac{\partial A}{\partial x}\right)$$

$$\frac{\partial A^{-1}}{\partial x} = -A^{-1} \frac{\partial A}{\partial x} A^{-1}$$

非線形連立方程式。通常の方法で解く。  
pythonならscipy.optimize.root関数。

# LASSOの場合

$$\min_{\mathbf{x}} \left\{ \frac{1}{2} \|\mathbf{y} - A\mathbf{x}\|_2^2 + \lambda \|\mathbf{x}\|_1 \right\}$$

LASSOはベイズ統計の観点では  
MAP推定 (maximum posterior)

$$\max_{\mathbf{x}} p(\mathbf{x}|\mathbf{y}, \lambda) \quad \text{事後確率分布}$$

$$p(\mathbf{x}|\mathbf{y}, \lambda) \propto p(\mathbf{y}|\mathbf{x})p(\mathbf{x}|\lambda)$$

尤度関数   事前確率

$$p(\mathbf{y}|\mathbf{x}) \propto e^{-\frac{1}{2} \|\mathbf{y} - A\mathbf{x}\|_2^2}$$

$$p(\mathbf{x}|\lambda) \propto e^{-\lambda \|\mathbf{x}\|_1}$$

改善するなら、

- ① まず  $\mathbf{x}$  について事後確率分布を考慮  
( $\mathbf{x}$  の揺らぎを考慮) して、 $\mathbf{x}$  について  
周辺化
- ② 次に  $\lambda$  について周辺化、あるいは  $\lambda$  に  
関する周辺化尤度を最大化

LASSOの数値解法は難しいので、①が  
実行困難。

よって、数値実験的な方法で  $\lambda$  の値を  
決定する。

# エルボー法 (Elbow method)

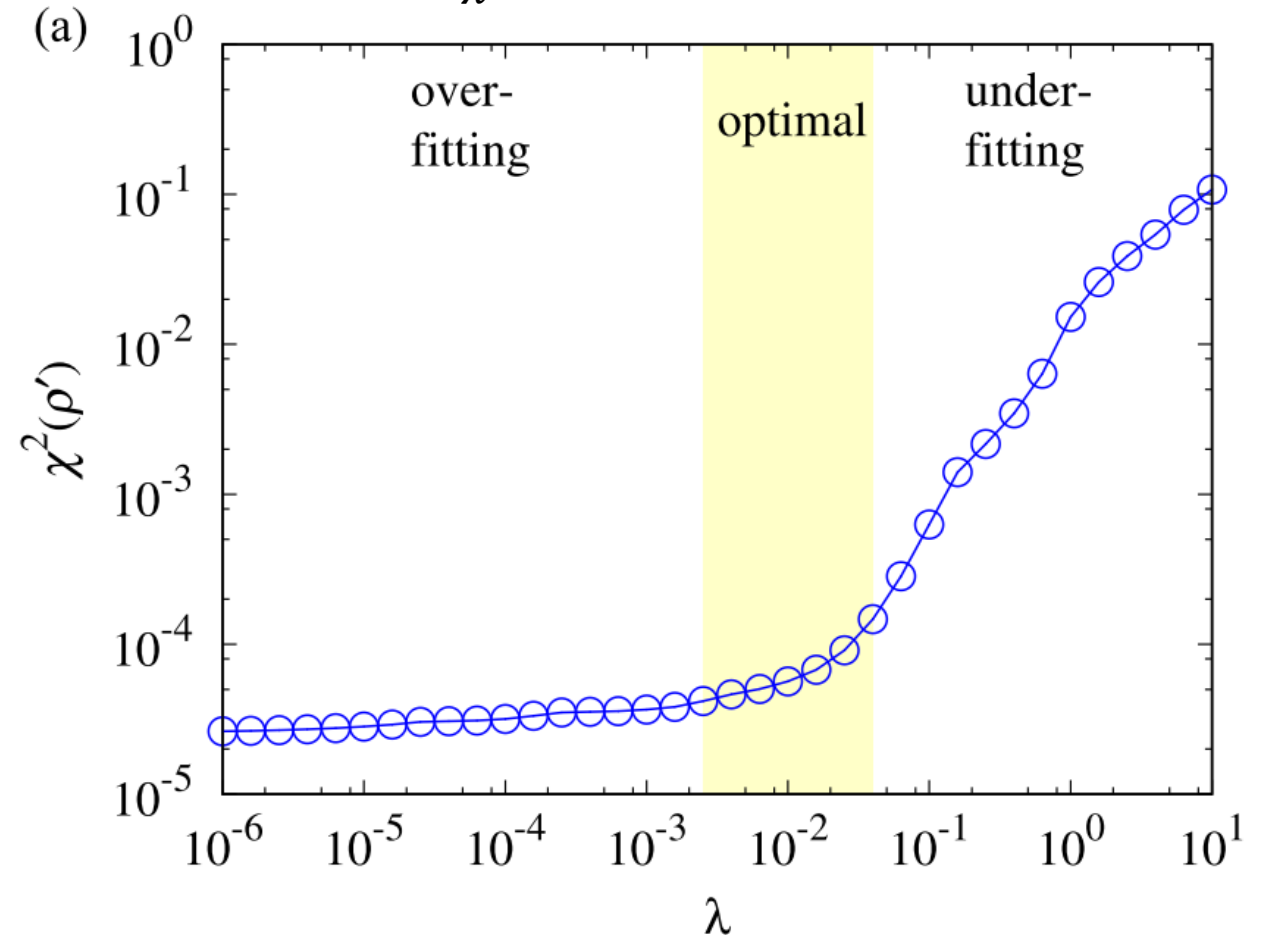
二乗誤差  $\chi^2$  vs 正則化パラメータ  $\lambda$  のグラフ（右図）で、**折れ曲がり**（肘）のあたりで適当な解が得られるという経験則

直感的な理解：

$\lambda > \lambda_{\text{opt}}$  では、 $\mathbf{x}$  の成分のひとつひとつが大きな寄与  $\rightarrow \chi^2$  が  $\lambda$  に敏感に反応

$\lambda < \lambda_{\text{opt}}$  では、 $\mathbf{x}$  の成分のひとつひとつが小さな寄与  $\rightarrow \chi^2$  が  $\lambda$  に鈍感

LASSOにおける  
二乗誤差  $\chi^2$  vs 正則化パラメータ  $\lambda$  の例

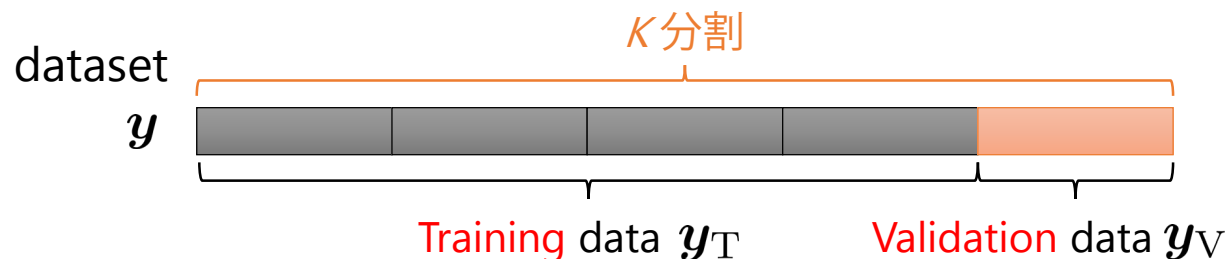


JO et al. JPSJ 2020

# 交差検証法 (Cross Validation; CV)

以下の手順により最も汎化性能 (generalization) が高い解を選ぶ

- ① 全データを Training data と Validation data に分ける



- ② Training data を使って解  $x^*$  を求める

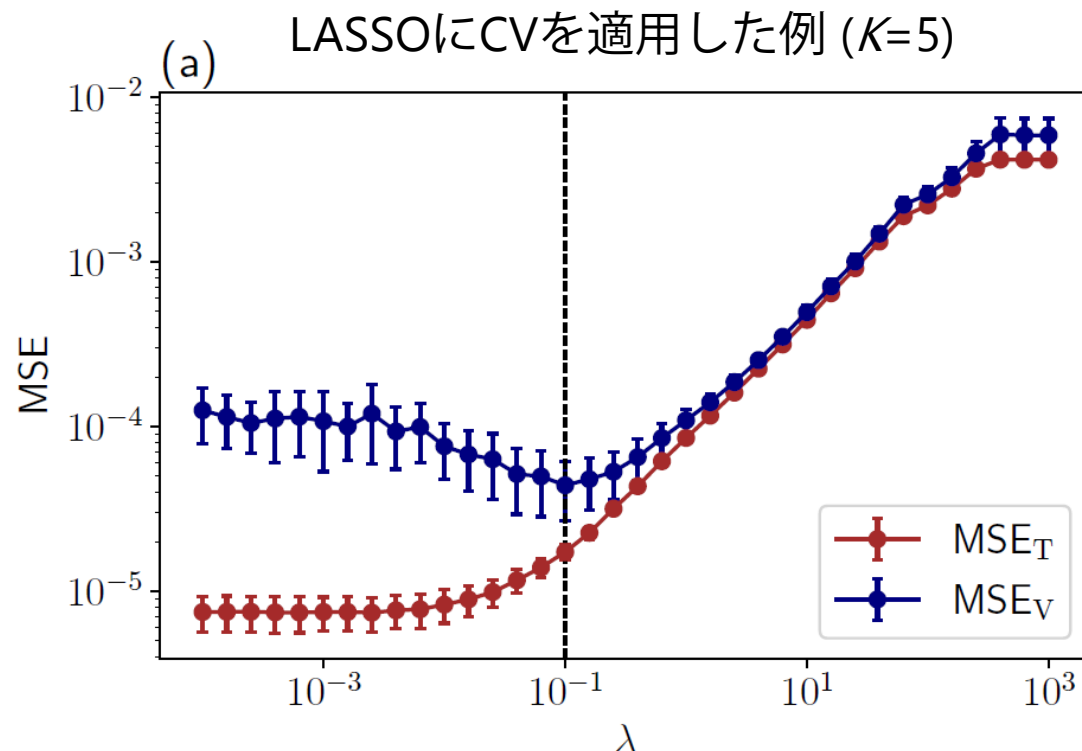
Training error  $\text{MSE}_T = \frac{1}{M_T} \|y_T - \mathcal{P}_T A x^*\|_2^2$  LASSO の場合

- ③ Validation data を使って解  $x^*$  の汎化性能を検証

Validation error  $\text{MSE}_V = \frac{1}{M_V} \|y_V - \mathcal{P}_V A x^*\|_2^2$

- ④  $K$  通りの組み合わせで②,③を行い  $\text{MSE}_V$  を平均化する

$$\text{CVE} = \frac{1}{K} \sum_{(y_T, y_V)} \text{MSE}_V$$



$\text{MSE}_T$  は正則化  $\lambda$  に対して単調増加  
 $\text{MSE}_V$  は極小をとる  $\rightarrow$  最も汎化性能が高い

# 交差検証法の種類

- K-fold (K-fold CV)

$K$ が大きいほど正確。ただし、与えられた $\lambda$ に対して最適化問題を $K$ 回解く必要があるため、 $K$ に比例して計算量が増える。 $K=5\sim 10$ くらいが実用的。

- Leave-one-out CV (LOOCV)

データを一つだけ検証用に残す(K-fold CVで「 $K = \text{data数}$ 」に相当)。最も計算量が多い。

- 近似公式 (小淵・樺島公式)

LOOCVを一度の最適化だけで評価する近似公式。LASSOやベイズ線形回帰など、それぞれに近似公式が導出されている。

T. Obuchi, Y. Kabashima, J. Stat. Mech. (2016) 053304