情報認識
「最大事後確率推定法と経験ベイズ法」

- 担当教員： 杉山　将（計算工学専攻）
- 居室： W8E-505
- 電子メール: sugi@cs.titech.ac.jp

ベイズ推定法の近似

$$\hat{p}(x) = \int_{\Theta} q(x;\theta) \frac{\prod_{i=1}^{n} q(x_i;\theta)p(\theta)}{\int_{\Theta} \prod_{i=1}^{n} q(x_i;\theta')p(\theta')d\theta'} d\theta$$

- ラプラス近似を用いれば，比較的簡単にベイズ推定法が計算できる．
- しかし，ヘシアン行列を計算するのが面倒．
- もっと計算を簡単にできないか？

最大事後確率推定法

- **最大事後確率推定法** (maximum a posteriori estimation method, MAP**推定法**)：**パラメータに関して積分するのをやめて，事後確率を最大にするパラメータ1点を用いる**

$$\hat{p}(x) = q(x; \hat{\theta}_{MAP})$$

$$\hat{\theta}_{MAP} = \arg\max_{\theta} p(\theta \mid x_1, x_2, \ldots, x_n)$$

- **ベイズの定理より**

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \left[\sum_{i=1}^{n} \log q(x_i; \theta) + \log p(\theta) \right]$$

最大事後確率推定法と最尤推定法

- MAP推定法はベイズ推定の近似解法.
- しかし，代表パラメータ $\hat{\theta}_{MAP}$ で分布を推定するため，ベイズ推定よりはむしろ最尤推定に近い.
- MAP推定法は，対数尤度に対数事前確率を加えたものを最大にする.
- そのためMAP法は，**罰則付き最尤推定法** (penalized MLE)とも呼ばれる.

$$\hat{\theta}_{MAP} = \arg\max_{\theta} \left(\underbrace{\sum_{i=1}^{n} \log q(x_i;\theta)}_{\text{対数尤度}} + \underbrace{\log p(\theta)}_{\text{対数事前確率}} \right)$$

演習

- 以下の設定でMAP推定量と最尤推定量を求めよ.
  - モデル：分散1のガウスモデル

$$q(x;\mu)=\frac{1}{\sqrt{2\pi}}\exp\left(-\frac{(x-\mu)^2}{2}\right)$$

  - 標本：　$\{x_i\}_{i=1}^n,\ x_i\in R$
  - 事前確率：ガウス分布　$p(\mu)=\frac{1}{\sqrt{2\pi\beta^2}}\exp\left(-\frac{\mu^2}{2\beta^2}\right)$

$\beta>0$

(小さい μ の方が出やすい)

- MAP推定量と最尤推定量の関係を調べよ.

演習の解答

■**罰則付き対数尤度**：

$$PL(\mu) = \sum_{i=1}^{n} \log q(x_i;\mu) + \log p(\mu)$$

$$= -\frac{n}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{n}(x_i-\mu)^2 - \frac{1}{2}\log 2\pi\beta^2 - \frac{\mu^2}{2\beta^2}$$

■**罰則付き尤度方程式**：

$$\frac{\partial}{\partial\mu}PL(\mu) = \sum_{i=1}^{n}(x_i-\mu) - \frac{\mu}{\beta^2} = 0 \quad \Rightarrow \quad \hat{\mu}_{MAP} = \frac{1}{n+\beta^{-2}}\sum_{i=1}^{n} x_i$$

■**通常の最尤推定量は** $\hat{\mu}_{MLE} = \frac{1}{n}\sum_{i=1}^{n} x_i$

■$|\hat{\mu}_{MAP}|$ **の方が** $|\hat{\mu}_{MLE}|$ **よりも小さい**： $|\hat{\mu}_{MAP}| = \frac{n}{n+\beta^{-2}}|\hat{\mu}_{MLE}| < |\hat{\mu}_{MLE}|$

事前確率の役割

- 最尤推定法は訓練標本が少ないとき精度がよくないことがある.
- MAP推定法では事前確率の大きい方に解が引っ張られるので, 訓練標本が少ない場合でも精度がよいことがある.
- 事前確率に関する知識がない場合, 事前確率を自分で設定する必要がある.
- 事前確率によって推定結果が変わるため, 客観的な方法で事前確率を設定しないと意味のある推定結果が得られない.

事前確率の設定

- あるパラメータ β で事前分布が制御できる場合を考える： $p(\theta;\beta)$
- 普通のパラメータ θ と区別するため，β を**超パラメータ**(hyper-parameter)とよぶ.
- 事前分布の設定の仕方
  - データから事前分布を定める (**経験ベイズ法**)
  - 事前分布の事前分布 (超事前分布) を考えて，平均を取る

経験ベイズ法

- **経験ベイズ法**(empirical Bayes method)：手元にある訓練標本が最も生起されやすいように超パラメータ β の値を設定.
- 訓練標本 $\{x_i\}_{i=1}^n$ が生起する確率：

$$p(x_1, x_2, \ldots, x_n; \beta) = \int_\Theta p(x_1, x_2, \ldots, x_n \mid \theta) p(\theta; \beta) d\theta$$

$$= \int_\Theta \prod_{i=1}^n q(x_i; \theta) p(\theta; \beta) d\theta$$

- **周辺尤度**(marginal likelihood)：これを β の関数と見たもの.

$$ML(\beta) = \int_\Theta \prod_{i=1}^n q(x_i; \theta) p(\theta; \beta) d\theta$$

経験ベイズ法 (続き)

- 周辺尤度を最大にするように β を決定.

$$\beta_{EB} = \arg\max_{\beta} ML(\beta)$$

$$ML(\beta) = \int_{\Theta} \prod_{i=1}^{n} q(x_i;\theta) p(\theta;\beta) d\theta$$

- この方法は**第 II 種最尤推定法**(type II maximum likelihood estimation method)ともよばれる.
- 周辺尤度は, **証拠**(evidence), **確率的複雑さ**(stochastic complexity), あるいは**自由エネルギー**(free energy)ともよばれる.

周辺尤度の計算

- 周辺尤度は積分を含んでいるため，計算が大変なので，ラプラス近似を使う．
- 実際には対数周辺尤度の方が計算しやすい．

$$\log ML(\beta) \approx g(\hat{\theta}_{MAP};\beta) + \frac{\dim\theta}{2}\log 2\pi - \frac{1}{2}\log|-\beta|$$

$$\hat{\theta}_{MAP} = \arg\max_{\theta}\left[g(\theta;\beta)\right]$$

$$H_{i,j}^{(\beta)} = \frac{\partial^2}{\partial\theta_i\partial\theta_j}\left[g(\theta;\beta)\right]_{\theta=\hat{\theta}_{MAP}}$$

$$g(\theta;\beta) = \sum_{k=1}^{n}\log q(x_k;\theta) + \log p(\theta;\beta)$$

- 訓練標本数が大きい時，この近似は精度がよい．

最適な超パラメータの探索

$$\beta_{EB} = \arg\max_{\beta} \int_{\Theta} p(x_1, x_2, \ldots, x_n \mid \theta)\, p(\theta; \beta)\, d\theta$$

- 周辺尤度の積分はラプラス近似で簡単に計算できるが，それを最大にする β_{EB} を解析的に求めるのは困難．
- 以下のように探索する．
  1. β の候補をいくつか用意
  2. それらに対して周辺尤度を計算
  3. その中で周辺尤度が最大のものを選ぶ

周辺尤度を用いたモデル選択

- 最尤推定のときと同様に，MAP推定でもモデルを適切に選ぶ必要がある．
- 超パラメータの設定と同様に，全てのモデルの候補に対して周辺尤度を計算し，周辺尤度最大のモデルを選べばよい．

ラプラス近似の更なる近似

- 訓練標本が十分多いことを仮定すると，

$$\sum_{i=1}^{n} \log q(x_i;\hat{\theta}_{MAP}) = O(n) \qquad \frac{\dim\theta}{2}\log 2\pi = O(1)$$

$$\log p(\hat{\theta}_{MAP};\beta) = O(1) \qquad \frac{1}{2}\log|H| = O(\log n)$$

$$\hat{\theta}_{MAP} \approx \hat{\theta}_{ML} = O(1) \qquad \frac{1}{2}\log|H| \approx \frac{\dim\theta}{2}\log n$$

- $O(1)$ の項を無視することにすれば

$$\log\int_{\Theta}\prod_{i=1}^{n} q(x_i;\theta)p(\theta;\beta)d\theta$$

$$\approx \sum_{i=1}^{n}\log q(x_i;\hat{\theta}_{ML}) - \frac{\dim\theta}{2}\log n$$

ベイズ情報量規準

- **ベイズ情報量規準**(Bayesian information criterion)：

$$BIC = -\sum_{i=1}^{n} \log q(x_i; \hat{\theta}_{ML}) + \frac{1}{2} \dim \theta \log n$$

- BIC**はもはや超パラメータ** β **に依らない.**
- **従って, 超パラメータの決定に**BIC**を用いることは出来ない.**
- **パラメトリックモデルの選択には使える.**

AIC vs. BIC

- AICとBICは似た形：

$$AIC = -\sum_{i=1}^{n} \log q(x_i; \hat{\theta}_{ML}) + \dim \theta$$

$$BIC = -\sum_{i=1}^{n} \log q(x_i; \hat{\theta}_{ML}) + \frac{1}{2} \dim \theta \log n$$

- BICの方が複雑なモデルに対する罰則が強いため，単純なモデルを好む傾向がある．
- どちらがよいかは場合と立場による．

まとめ

- 最大事後確率推定法 (MAP法) ：ベイズ推定の積分を一点で近似する
- MAP法はベイズ推定よりもむしろ最尤推定に近い (罰則付き最尤推定法とも呼ばれる)
- ベイズの枠組みでは，周辺尤度を最大にするように超パラメータやモデルを決定するのが自然.
- これは，今手持ちのデータが最も生起されやすいように超パラメータやモデルを決定することに対応する.
- 周辺尤度は，ラプラス近似やBIC近似により効率よく計算できる.

小レポート (第12回)

■ **モデル**: $q(x;\mu) = \frac{1}{\sqrt{2\pi}} \exp\left(-\frac{(x-\mu)^2}{2} \right)$

■ **標本**: $\{x_i\}_{i=1}^n,\ x_i \in R$

■ **事前確率**: $p(\mu;\beta) = \frac{1}{\sqrt{2\pi\beta^2}} \exp\left(-\frac{\mu^2}{2\beta^2} \right)$ $\beta > 0$

1. **上記の設定のもと, 対数周辺尤度のラプラス近似が次式で与えられることを示せ.**

$$\log ML(\beta) \approx -\frac{n}{2}\log 2\pi - \frac{1}{2}\sum_{i=1}^{n}(x_i - \hat{\mu}_{MAP})^2 - \frac{\hat{\mu}_{MAP}{}^2}{2\beta^2} - \frac{1}{2}\log(n\beta^2 + 1)$$

$$\hat{\mu}_{MAP} = \frac{1}{n + \beta^{-2}} \sum_{i=1}^{n} x_i$$

小レポート (続き)

2. Octaveなどを用いた実験:

   平均0．5分散1の正規分布からn個標本を生成し，1．のモデルに対して経験ベイズ法で β を決定せよ．具体的には，β の候補を幾つか用意し，それぞれに対して対数周辺尤度をラプラス近似で求め，対数周辺尤度の最も大きいものを選べ．

   そうして得られた β を用いてMAP推定法で μ を推定せよ．

   上記の実験を真の平均や標本数を変えて実験を行い，

   - 経験ベイズ法＋MAP推定法
   - 最尤推定法

   の精度を比較せよ．

Octaveのサンプルプログラム

ex12.m

```
clear all
n=20; mu=0.5; sigma=1;
xx=sigma*randn(n,1)+mu;
betas=[0.01:0.01:3]; b=length(betas);

mu_MLE=mean(xx);
sigma_MLE=std(xx,1);
for i=1:b
  beta=betas(i);
  mu_MAP(i)=sum(xx)/(n+beta.^(-2));
  logML(i)=-n/2*log(2*pi) ...
              -sum((xx-mu_MAP(i)).^2)/2 ...
              -mu_MAP(i)^2/(2*beta^2) ...
              -log(n*beta^2+1);
end
[dummy,c]=max(logML);a
```

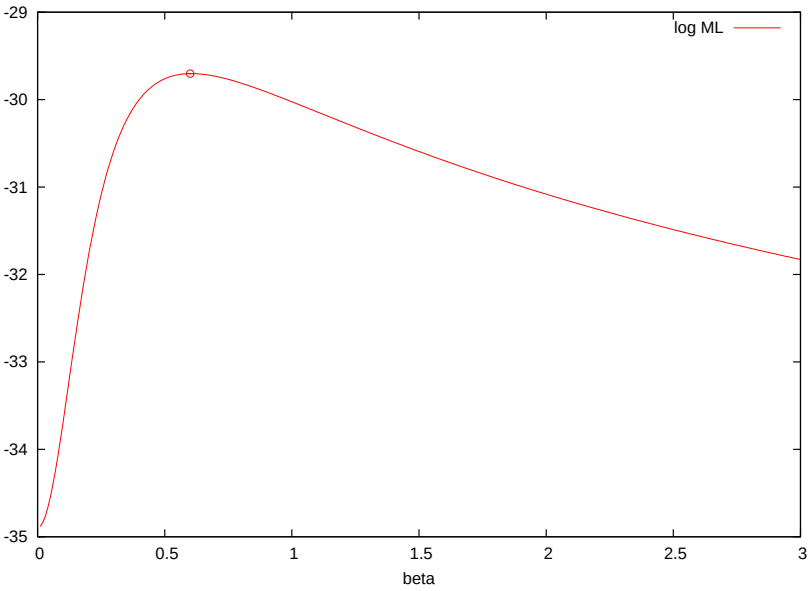
(右上に続く)

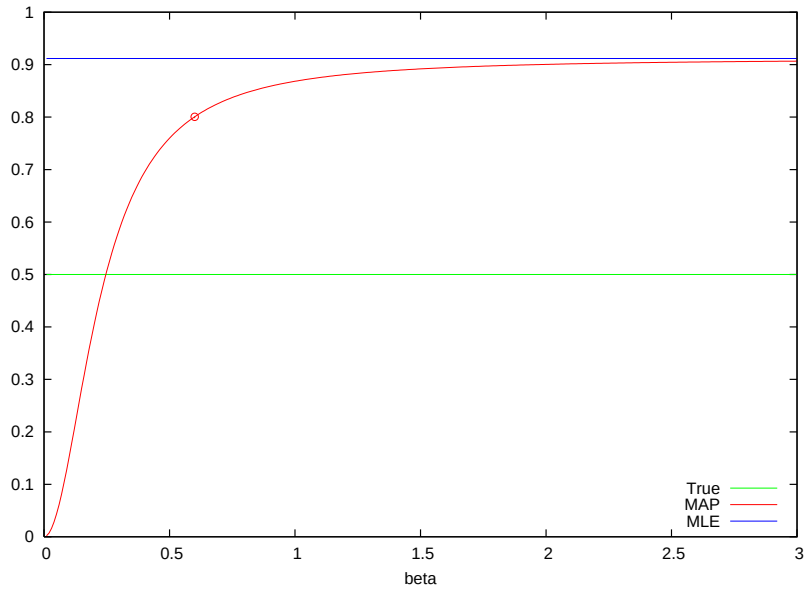
(左下から続き)

```
figure(1); clf; hold on;
plot(betas,logML,'r-')
plot(betas(c),logML(c),'@16')
xlabel('beta')
legend('log ML')
print -deps logML.eps

figure(2); clf; hold on;
plot(betas,mu*ones(1,b),‘g-')
plot(betas,mu_MAP,'r-')
plot(betas,mu_MLE*ones(1, b),'b-')
plot(betas(c),mu_MAP(c),'@16')
xlabel('beta')
legend('True','MAP','MLE',4)
print -deps MAP.eps
```

実行例


log ML
-29
-30
-31
-32
-33
-34
-35
0
0.5
1
1.5
2
2.5
3
beta


1
0.9
0.8
0.7
0.6
0.5
0.4
0.3
0.2
0.1
0
0
0.5
1
1.5
2
2.5
3
beta
True
MAP
MLE