10.2 変分推論-混合ガウス分布-

平成 28 年 9 月 11 日

概 要

PRML の「10.2 例:変分混合ガウス分布」についての実装と考察

目 次

1 問題設定

混合ガウス分布に対して, 分解による変分近似を行う.

2 アルゴリズム

同一の混合ガウス分布から独立に発生したと仮定するデータ集合 $D = \{x_1, \ldots, x_n\}$ が与えられたとき, 元の混合ガウス分布の平均 $\boldsymbol{\mu}$ と精度 $\boldsymbol{\Lambda}$ と混合比 $\boldsymbol{\pi}$ の事後分布を求めることが目標である.
まず, 観測変数を X, 潜在変数を Z と分ける.
混合比 $\boldsymbol{\pi}$ が与えられたときの Z の条件付き分布は

$$p(Z|\boldsymbol{\pi}) = \prod_{n=1}^{N}\prod_{k=1}^{K}\pi_k^{z_{nk}} \quad (10.37)$$

潜在変数と混合要素のパラメータが与えられたときの X の条件付き分布は

$$p(X|Z,\boldsymbol{\mu},\boldsymbol{\Lambda}) = \prod_{n=1}^{N}\prod_{k=1}^{K}N(\mathbf{x}_n|\boldsymbol{\mu}_k,\Lambda_k^{-1})^{z_{nk}} \quad (10.38)$$

となる.
また, パラメータの事前分布は, ディリクレ分布とガウス-ウィシャート分布を選び

$$p(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}_0) = C(\boldsymbol{\alpha}_0)\prod_{k=1}^{K}\pi_k^{\alpha_0-1} \quad (10.39)$$

$$p(\boldsymbol{\mu},\Lambda) = p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) = \prod_{k=1}^{K}N(\boldsymbol{\mu}_k|\mathbf{m}_0,(\beta_0\Lambda_k)^{-1})W(\Lambda_k|\mathbf{w}_0,\nu_0) \quad (10.40)$$

とする.
ここで, 変分ベイズ法を用いるために同時分布の分解を考える.

$$p(X,Z,\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) = p(X|Z,\boldsymbol{\mu},\boldsymbol{\Lambda})p(Z|\boldsymbol{\pi})p(\boldsymbol{\pi})p(\boldsymbol{\mu}|\boldsymbol{\Lambda})p(\boldsymbol{\Lambda}) \quad (10.41)$$

となる. そして, 潜在変数とパラメータに分解した変分近似は

$$q(Z,\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) = q(Z)q(\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}) \quad (10.42)$$

となるが, この各因子をそれぞれ推測したい. これには次の

$$\ln q_j^*(\boldsymbol{Z}_j) = E_{i\neq j}[\ln p(\boldsymbol{X},\boldsymbol{Z})] + const \quad (10.9)$$

を用いる.
まず, $\ln q(Z)$ については

$$\begin{aligned}
\ln q*(Z) &= E_{\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(X,Z,\boldsymbol{\pi},\boldsymbol{\mu},\boldsymbol{\Lambda})] + const \quad (10.43)\\
&= E_{\boldsymbol{\pi}}[\ln p(Z|\boldsymbol{\pi})] + E_{\boldsymbol{\mu},\boldsymbol{\Lambda}}[\ln p(X|Z,\boldsymbol{\mu},\boldsymbol{\Lambda})] + const \quad (10.44)\\
&= \sum_{n=1}^{N}\sum_{k=1}^{K}z_{nk}\ln\rho_{nk} + const \quad (10.45)
\end{aligned}$$

ただし,

$$\ln\rho_{nk} = E[\ln\pi_k] + \frac{1}{2}E[\ln|\Lambda_k|] - \frac{D}{2}\ln 2\pi - \frac{1}{2}E_{\boldsymbol{\mu}_k,\Lambda_k}[(\mathbf{x}_n-\boldsymbol{\mu}_k)^T\Lambda_k(\mathbf{x}_n-\boldsymbol{\mu}_k)] \quad (10.46)$$

となる. そして, 両辺の指数をとることによって (正規化も考慮し)

$$q*(Z) = \prod_{n=1}^{N}\prod_{k=1}^{K} r_{nk}^{z_{nk}} \quad (10.48)$$

ただし,

$$r_{nk} = \frac{\rho_{nk}}{\sum_j \rho_{nj}} \quad (10.49)$$

また, 同様にして $\ln q(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda})$ については

$$\ln q^*(\boldsymbol{\pi}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = \ln p(\boldsymbol{\pi}) + \sum_{k=1}^{K} \ln p(\boldsymbol{\mu}_k, \Lambda_k) + E_Z[\ln p(Z|\boldsymbol{\pi})] + \sum_{n=1}^{N}\sum_{k=1}^{K} E[z_n k] \ln N(\mathbf{x}_n|\boldsymbol{\pi}, \boldsymbol{\mu}_k, \Lambda_k^{-1}) + const$$

で, これは $\boldsymbol{\pi}$ だけの項と $\boldsymbol{\mu}$ と $\boldsymbol{\Lambda}$ だけの項に分解されることが分かり, これは変分事後分布が $q(\boldsymbol{\mu}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\mu})q(\boldsymbol{\mu}, \boldsymbol{\Lambda})$ と分解されることを表す.

$$q(\boldsymbol{\mu}, \boldsymbol{\mu}, \boldsymbol{\Lambda}) = q(\boldsymbol{\mu}) \prod_{k=1}^{K} q(\boldsymbol{\mu}_k, \Lambda_k) \quad (10.55)$$

まず, $\boldsymbol{\pi}$ に依存する項を探すと

$$\ln q^*(\boldsymbol{\pi}) = (\alpha_0 - 1)\sum_{k=1}^{K} \ln \pi_k + \sum_{k=1}^{K}\sum_{n=1}^{N} r_{nk} \ln \pi_k + const \quad (10.56)$$

となるが, 両辺の指数をとることで

$$q^*(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha}) \quad (10.57)$$

ただし,

$$\alpha_k = \alpha_0 + N_k \quad (10.58), \quad N_k = \sum_{n=1}^{N} r_{nk} \quad (10.51)$$

である.
また, $\boldsymbol{\mu}_k, \Lambda_k$ に依存する項を探すと

$$q^*(\boldsymbol{\mu}_k, \Lambda_k) = N(\boldsymbol{\mu}_k|\boldsymbol{m}_k, (\beta_k\Lambda_k)^{-1})W(\Lambda_k|W_k, \nu_k) \quad (10.59)$$

ただし,

$$\bar{\boldsymbol{x}_k} = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}\boldsymbol{x}_n \quad (10.52), \quad S_k = \frac{1}{N_k}\sum_{n=1}^{N} r_{nk}(\mathbf{x}_n - \bar{\mathbf{x}_k})(\mathbf{x}_n - \bar{\mathbf{x}_k})^T \quad (10.53)$$

$$\beta_k = \beta_0 + N_k \quad (10.60), \quad \mathbf{m}_k = \frac{1}{\beta_k}(\beta_0\mathbf{m}_0 + N_k\bar{\mathbf{x}_k}) \quad (10.61), \quad \nu_k = \nu_0 + N_k \quad (10.63)$$

$$W_k^{-1} = W_0^{-1} + N_k S_k + \frac{\beta_0 N_k}{\beta_0 + N_k}(\bar{\mathbf{x}_k} - \mathbf{m}_0)(\bar{\mathbf{x}_k} - \mathbf{m}_0)^T \quad (10.62)$$

となる.
そして, これらを計算するためには, $q^*(\boldsymbol{\pi})$ と $q^*(\boldsymbol{\mu}_k, \Lambda_k)$ のパラメータが相互に関係していることから, 繰り返しを必要とする.
また, これらを計算するには

$$E_{\boldsymbol{\mu}_k,\Lambda_k}[(\boldsymbol{x}_n - \boldsymbol{\mu}_k)^T\Lambda_k(\mathbf{x}_n - \boldsymbol{\mu}_k)] = D\beta_k^{-1} + \nu_k(\mathbf{x}_n - \boldsymbol{\mu}_k)^T W_k(\mathbf{x}_n - \boldsymbol{\mu}_k) \quad (10.64)$$

$$E[\ln|\Lambda_k|] = \sum_{i=1}^{D} \psi\left(\frac{\nu_k + 1 - i}{2}\right) + D\ln 2 + \ln|W_k| \quad (10.65)$$

$$E[\ln \pi_k] = \psi(\alpha_k) - \psi(\sum_k \alpha_k) \quad (10.66)$$

を用いることになる. ただし,$\psi(\cdot)$ はディガンマ関数である.

変分推論

1. 混合ガウス分布では, 混合比 $\boldsymbol{\pi}$, 平均 $\boldsymbol{\mu}$ と精度 $\boldsymbol{\Lambda}$ の共役事前分布にディリクレ分布, ガウス-ウィシャート分布を選ぶ.
2. 必要となるモーメント $E[z_{nk}] = r_{nk}$ を適当に初期化する.
3. 因子 $q^*(Z) = \prod_n \prod_k r_{nk}^{z_{nk}}$ のパラメータを推定する.
4. 因子 $q^*(\boldsymbol{\pi}) = Dir(\boldsymbol{\pi}|\boldsymbol{\alpha})$ のパラメータを推定する.
5. 因子 $q^*(\boldsymbol{\mu}_k, \Lambda_k) = N(\boldsymbol{\mu}_k|\mathbf{m}_k, (\beta_k\Lambda_k)^{-1})W(\Lambda_k|W_k, \nu_k)$ のパラメータを推定する.
6. 3,4,5 を繰り返す.

3 コード

ガウス分布に対する変分近似のコード (variation_mixture_of_gauss.py).

```
def gauss(x,mu,lam):
  return sqrt(1/2/pi)*exp(-dot(x-mu,dot(lam,x-mu))/2)

def digamma(x):
  return (gam(x+10**-2)-gam(x))*10**2

"""データセットの作成"""
dataset="dataset.csv"
data=sp.genfromtxt(dataset,delimiter=",")
D=2
K=4
for N in [10,50,100,200,500,1000]:
  x=data[:N,:D]
  t=data[:N,D:]
  """変分近似"""
  #超パラメータ(決め方がわからない)
  alpha0=1
  beta0=1
  m0=np.zeros(D)
  W0=np.identity(D)
  nu0=2

  #モーメント
  rho=rd.rand(N,K)

  r=rho
  for n in range(N):
    r[n,:]=rho[n,:]/np.sum(rho[n,:])

  Nk=np.zeros(K)
  Nk=[np.sum(r[:,k]) for k in range(K)]

  Xk=np.zeros((K,D))
```

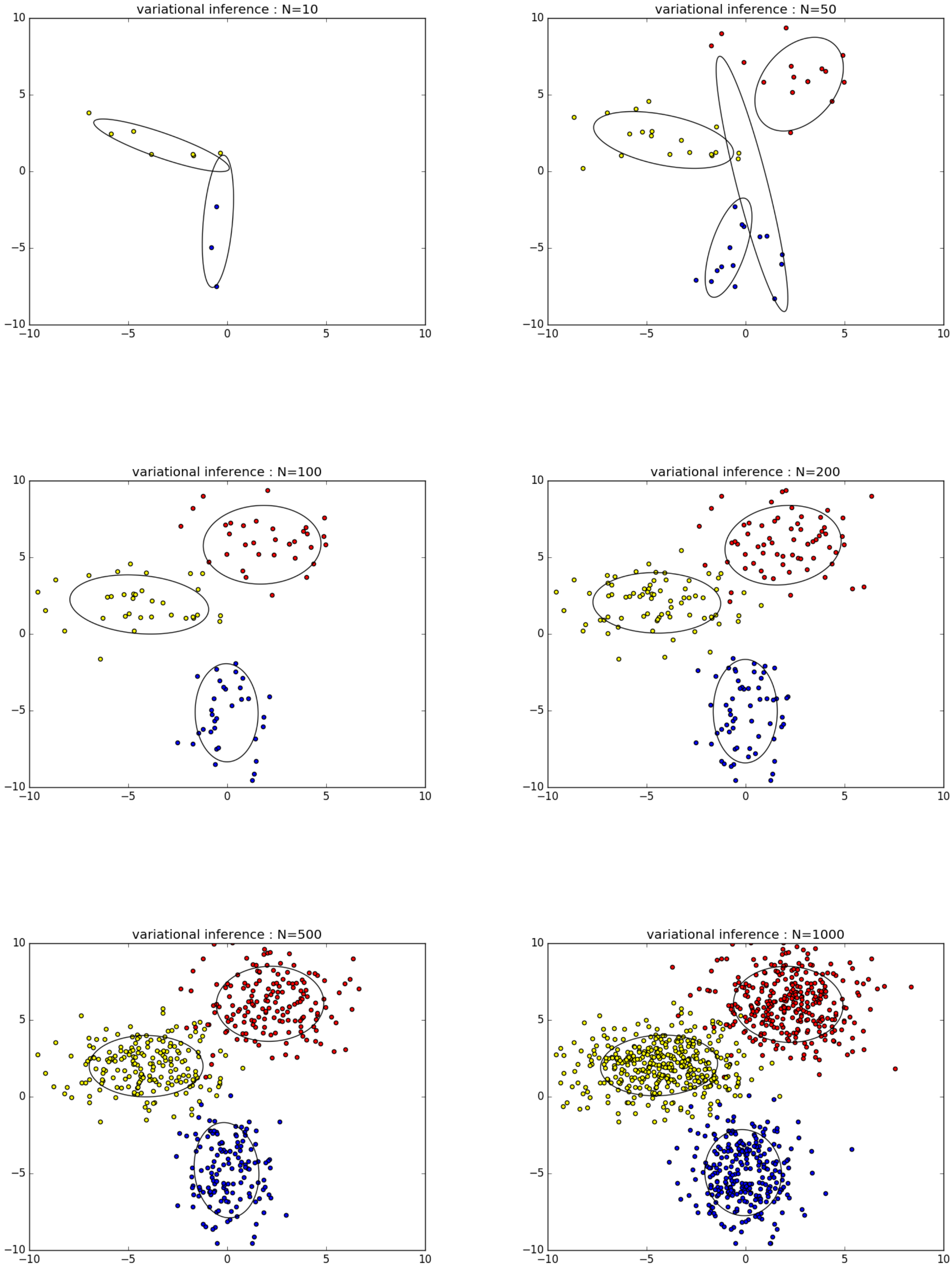
```
for k in range(K):
  for n in range(N):
    Xk[k,:]+=r[n,k]*x[n,:]
  Xk[k,:]/=Nk[k]

Sk=np.zeros((K,D,D))
for k in range(K):
  for n in range(N):
    Sk[k,:,:]+=r[n,k]*outer(x[n,:]-Xk[k,:],x[n,:]-Xk[k,:])
  Sk[k,:,:]/=Nk[k]

alphak=np.zeros(K)
for k in range(K):
  alphak[k]=alpha0+Nk[k]

betak=np.zeros(K)
for k in range(K):
  betak[k]=beta0+Nk[k]

mk=np.zeros((K,D))
for k in range(K):
  mk[k,:]=(beta0*m0[:]+Nk[k]*Xk[k,:])/betak[k]

Wk=np.zeros((K,D,D))
for k in range(K):
  Wk[k,:,:]=inv(inv(W0)+Nk[k]*Sk[k,:,:]+beta0*Nk[k]/
            (beta0+Nk[k])*outer(Xk[k,:]-m0[:],Xk[k,:]-m0[:]))

nuk=np.zeros(K)
for k in range(K):
  nuk[k]=nu0+Nk[k]

diff=10**6
while diff>=10**-3:
  diff=0
  tmp=r
  for n in range(N):
    for k in range(K):
      tmp1=D/betak[k]+nuk[k]*dot(x[n,:]-mk[k,:],dot(Wk[k,:,:],x[n,:]-mk[k,:]))
      tmp2=D*np.log(2)+np.log(det(Wk[k,:,:]))
      for i in range(D):
        tmp2+=digam((nuk[k]-i)/2)
      tmp3=digam(alphak[k])-digam(np.sum(alphak[:]))
      rho[n,k]=np.exp(tmp3+tmp2/2-D/2*np.log(2*pi)-tmp1/2)

  r=rho
  for n in range(N):
    r[n,:]=rho[n,:]/np.sum(rho[n,:])
  diff+=norm(tmp-r)

  tmp=Nk
  Nk=np.zeros(K)
  for k in range(K):
    Nk[k]=np.sum(r[:,k])
  diff+=norm(tmp-Nk)

  tmp=Xk
  Xk=np.zeros((K,D))
  for k in range(K):
    for n in range(N):
      Xk[k,:]+=r[n,k]*x[n,:]
    Xk[k,:]/=Nk[k]
  diff+=norm(tmp-Xk)
```

```
    tmp=Sk
    Sk=np.zeros((K,D,D))
    for k in range(K):
      for n in range(N):
        Sk[k,:,:]+=r[n,k]*outer(x[n,:]-Xk[k,:],x[n,:]-Xk[k,:])
      Sk[k,:,:]/=Nk[k]
    for k in range(K):
      diff+=norm(tmp[k,:,:]-Sk[k,:,:])

    tmp=alphak
    alphak=np.zeros(K)
    for k in range(K):
      alphak[k]=alpha0+Nk[k]
    diff+=norm(tmp-alphak)

    tmp=betak
    betak=np.zeros(K)
    for k in range(K):
      betak[k]=beta0+Nk[k]
    diff+=norm(tmp-betak)

    tmp=mk
    for k in range(K):
      mk[k,:]=(beta0*m0[:]+Nk[k]*Xk[k,:])/betak[k]
    diff+=norm(tmp-mk)

    tmp=Wk
    for k in range(K):
      Wk[k,:,:]=inv(inv(W0)+Nk[k]*Sk[k,:,:]+beta0*Nk[k]/
                (beta0+Nk[k])*outer(Xk[k,:]-m0[:],Xk[k,:]-m0[:]))
    for k in range(K):
      diff+=norm(tmp[k,:,:]-Wk[k,:,:])

    tmp=nuk
    for k in range(K):
      nuk[k]=nu0+Nk[k]
    diff+=norm(tmp-nuk)
```

4 結果

混合要素数 3 つの混合ガウス分布に対して $K = 4$ 混合の変分ベイズ混合ガウスモデルを適応した結果. ここでは, 混合比 0.1 以下の混合要素についてはプロットしていない.

図 1: $N = 10, 50, 100, 200, 500, 1000$



5　まとめ

N が小さいとき, 混合比の推定がうまくいかないことがある. また, N が大きくなるにつれて, 正確になっていくのがわかる. 過学習が発生していなくて, 混合要素数の推定もできて便利.

ただ, アルゴリズムの部分は混合ガウス分布以外には適応できず, 頭で考えなければならない部分が大きいと思った.
また, 超パラメータについても考える必要がある.