

PBPの学習の流れ

パラメータの初期化

各ファイルの__init__()

P.130 1-8行目。
Prior.pyで各種変数初期化。
ただし、 w と \tilde{f} に関して、
Supplementary material 6の(33),(34)で
初期化。実用上は nat がつくもので初期化。

実際の学習

Pbp.py do_pbp()

do_first_pass()

refine_prior()

一定回数学習

W, γ_y のパラメータの学習
各教師データごとにパラメータを更新
1, 順伝播で尤度を計算。
計算式はpbp.pyの__init__で設定。
(Network.pyのlogZ_Z1_Z2で(5.65)が、
output_probabilisticと呼ばれる、
network_layer.pyのoutput_probabilisticで
(5.66)-(5.73)を利用している。
2, それを微分。
Pbp.pyのadf_updateで微分、更新。
(実際はnewwork.pyのgenerate_updates)
微分はtheanoを使ってやっているので
逆伝播がわかりにくい。
おそらく、c版では
supplementary material 1の(1)-(22)を
利用していると思われる。
(21),(22)が逆伝播で求めたい微分。
(21),(22)の δ が逆伝搬している情報。

W, γ_w のパラメータの学習
(正則化のようなことを行っているように思う。)
5.2.5.3の8行目から。
実際はEP法を使っている。本では
インデックスがないことに注意。
Supplementary materialの(35)-(39)参照。
途中にあるように、その間で(5.59)(5.60)
(5.61)(5.62)を利用する。

PBPの学習の流れ

パラメータの初期化

各ファイルの__init__()

実際の学習

Pbp.py do_pbp()

do_first_pass()

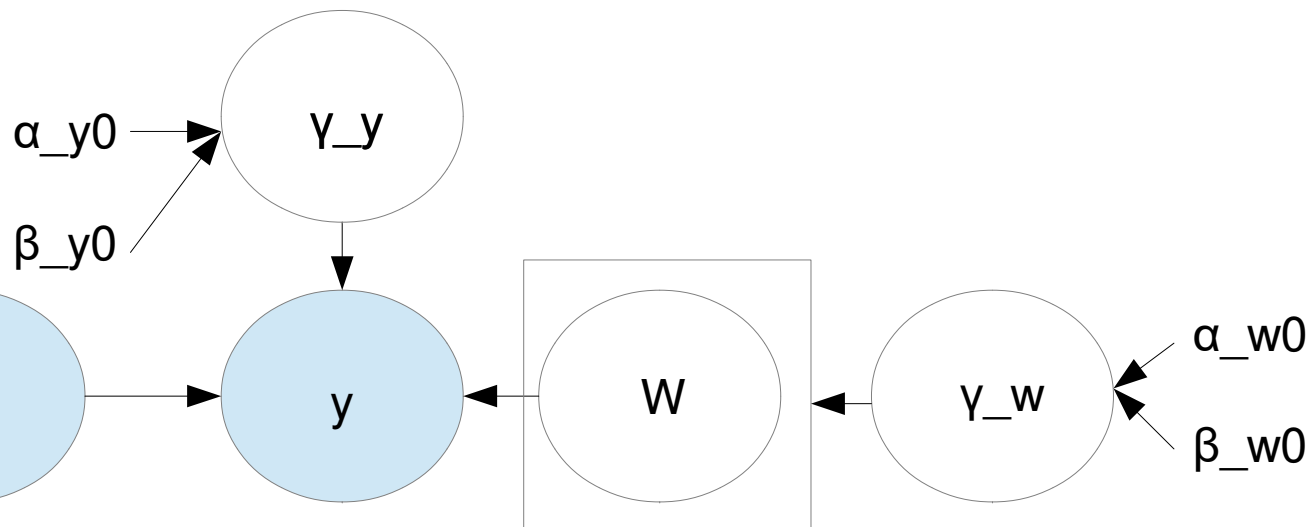
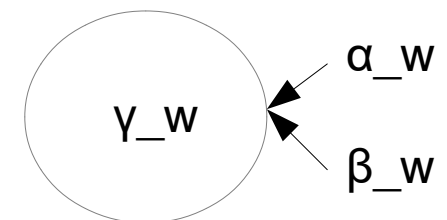
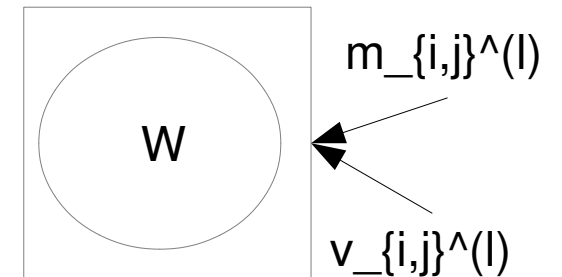
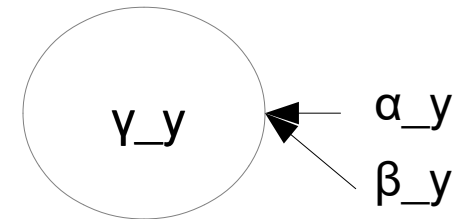
W, γ_y のパラメータの学習

refine_prior()

W, γ_w のパラメータの学習

一定回数学習

$q(W, \gamma_y, \gamma_w)$
(各変数が独立)



4章

5章

EP法	ADF	W, γ_y の学習	W, γ_w の学習
θ $f_n(\theta)$ $(n \geq 0)$ これに対して、 $p(\theta) = \prod f_n(\theta)$	θ 尤度として 書けるとき $f_0(\theta) = p(\theta)$ $f_i(\theta) \propto p(D_i \theta)$ これに対して、 $q(\theta) = \prod \tilde{f}_i(\theta)$	$W, \gamma_y(, \gamma_w)$ γ_w は積分して定数になる $f_0(\theta) = q(W, \gamma_w, \gamma_y)$ $f_{\{W, \gamma_y\}}(W, \gamma_y) =$ $p(\gamma_w W, \gamma_y)$ $\tilde{f}_W(\theta) = q(W)$ $= N(W)$ $\tilde{f}_{\gamma_y}(\theta) = q(\gamma_y)$ $= N(\gamma_y)$	$W, \gamma_w(, \gamma_y)$ γ_y は積分して定数になる $f_{\gamma_w}(\theta) = p(\gamma_w)$ $f_{\gamma_y}(\theta) = p(\gamma_y)$ $f_{w_{i,j,l}}(\theta) = p(w_{i,j,l} 0, \gamma_w)$ $\tilde{f}_{\gamma_w} = q(\gamma_w)$ $= \text{Gam}(\gamma_w)$ $\tilde{f}_{\gamma_y} = q(\gamma_y)$ $= \text{Gam}(\gamma_y)$ $w_{i,j,l}$ を含む部分について、 $\tilde{f}_{w_{i,j,l}}(\theta)$ $= N(w_{i,j,l}) \text{Gam}(\gamma_w)$