CS688 Homework 05

Sourabh Kulkarni | 29572060

Problem 01

1.1 The joint distribution can be written as follows, using first principles:

$P_\theta(X = x, Z = z) = P_\theta(X = x|Z = z) \cdot P_\theta(Z = z)$

where,

$$P_\theta(\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{z}; 0, I)$$
$$P_\theta(\mathbf{X} = \mathbf{x}|\mathbf{Z} = \mathbf{z}) = \mathcal{N}(\mathbf{x}; \mathbf{W}\mathbf{z}, \Psi)|$$

product of gaussians is also a gaussian,

$P_\theta(X = x, Z = z) = \mathcal{N}([x; z], \mu_{X:Z}, \Sigma_{X;Z})$

where,

$\mu_{x:z} = \begin{pmatrix}\mu_x \\ \mu_z\end{pmatrix}$, and $\Sigma_{x;z} = \begin{bmatrix}\Sigma_{XX} & \Sigma_{XZ} \\ \Sigma_{ZX} & \Sigma_{ZZ}\end{bmatrix}$

all these terms are now derived:

$x = Wz;\ \therefore\ \mu_x = W\mu_z = 0$

$\mu_{x:z} = \begin{pmatrix}0 \\ 0\end{pmatrix}$

$\Sigma_{XX} = cov(X) = Var(x_i) + cov(x_i, x_j)\ for\ all\ i \neq j$

$\therefore\ \Sigma_{XX} = \Psi + \mathrm{WW}^T$

$\Sigma_{XZ} = W, and\ \Sigma_{ZX} = \mathrm{W}^T\ as\ X = WZ$

$\Sigma_{ZZ} = I, given$

Aggregating all results, we have:

$P_\theta(X = x, Z = z) = \mathcal{N}([x; z], 0, \Sigma_{X;Z})$

where,

$\Sigma_{x;z} = \begin{bmatrix}\Psi + \mathrm{WW}^T & W \\ \mathrm{W}^T & I\end{bmatrix}$

1.2 The conditional of a joint multivariate gaussian is also a gaussian

$P_\theta(Z = z|X = x) = \mathcal{N}\big([z|x], \mu_{Z|X}, \Sigma_{Z|X}\big), where$

$\mu_{Z|X} = \mu_Z + \Sigma_{XZ}{\Sigma_{XX}}^{-1}(x - \mu_x)$

$= \mathrm{W}^T(\Psi + \mathrm{WW}^T)^{-1}x$

$\Sigma_{Z|X} = \Sigma_{ZZ} - \Sigma_{ZX}{\Sigma_{XX}}^{-1}\Sigma_{XZ}$

$= \mathrm{I} - \mathrm{W}^T(\Psi + \mathrm{WW}^T)^{-1}\mathrm{W}$

1.3

from 1.2 we have

$P_\theta(Z = z|X = x) = \mathcal{N}\big([z|x], \mu_{Z|X}, \Sigma_{Z|X}\big)$

We now derive m, S for

$Q_\phi(Z = z) = \mathcal{N}(z; m, S)$

From the KL divergence Wikipedia page, we have that the KL divergence of two multivariate gaussians with same dimensionality is given by:

$$D_{KL}\left(Q_\phi||P_\theta)\right) = \frac{1}{2}\Big(tr(\Sigma_{Z|X}{}^{-1}S)) + \left(\mu_{Z|X} - m\right)^T\Sigma_{Z|X}{}^{-1}(\mu_{Z|X} - m) - k + \log(\det(\Sigma_{Z|X})) - \log(\det(S))\Big)$$

where k is number of dimensions of Z. We know that the minimum value of the KL is 0. Optimal m, S should achieve that value. For that to happen 3 things need to be true

$$\left(\mu_{Z|X} - m\right)^T\Sigma_{Z|X}{}^{-1}\left(\mu_{Z|X} - m\right) = 0 \quad (1)$$

$$tr(\Sigma_{Z|X}{}^{-1}S)) = k \quad (2)$$

$$\log\left(\det\left(\Sigma_{Z|X}\right)\right) = \log(det(S)) \quad (3)$$

let's start with (1)

$$\left(\mu_{Z|X} - m\right)^T\Sigma_{Z|X}{}^{-1}\left(\mu_{Z|X} - m\right) = 0$$

$$\mu_{Z|X} - m = 0$$

$$m = \mu_{Z|X}$$

we have our optimal m:

$$m = \mathrm{W}^T(\Psi + \mathrm{WW}^T)^{-1}x$$

now let's obtain optimal S starting from the differentiation of the KL divergence expression w.r.t S. Using the result that the derivative of logdet(S) is $\frac{1}{S}$. Also, re-writing the $tr(\Sigma_{Z|X}{}^{-1}S))$term as sum of product of diagonal elements of $\Sigma_{Z|X}{}^{-1}$and S we get:

$$\frac{\delta D_{KL}\left(Q_\phi||P_\theta)\right)}{\delta S} = \left[-\frac{1}{S} + \Sigma_{Z|X}{}^{-1}\right]$$

equating to zero we get

$$S = \frac{1}{\Sigma_{Z|X}{}^{-1}}$$

Thus, the optimal diagonal elements in S are the reciprocals of the diagonal elements of $\Sigma_{Z|X}{}^{-1}$

Problem 3

3.1

Exact: **mean** = [0.66666667,0.5]

covariance = $\begin{bmatrix} 0.33333333 & 0 \\ 0 & 0.5 \end{bmatrix}$

BBSVI: **mean** = [0.66971193,0.49780578]

covariance = $\begin{bmatrix} 0.33954225 & 0 \\ 0 & 0.47624053 \end{bmatrix}$

3.2

Exact: **mean** = [0.5,0.5]

covariance = $\begin{bmatrix} 0.375 & -0.125 \\ -0.125 & 0.375 \end{bmatrix}$

BBSVI: **mean** = [0.50707745, 0.51860241]

covariance = $\begin{bmatrix} 0.34590475 & 0 \\ 0 & 0.32601725 \end{bmatrix}$

3.3 The variational approximation distribution has a covariance matrix limited to a diagonal matrix. This constraint limits its ability to approximate certain distributions.

In 3.1 the covariance matrix of the distribution to be approximated also happens to be diagonal, so the variational approximation distribution can obtain a close approximation. In 3.2, though, the covariance matrix of the distribution has some non-diagonal elements, which the variational approximation cannot capture due to its diagonal matrix constraint, which results in a poor approximation.

3.4 (discussed with Daniel Sam Fdo)

3.4.1 The parameters' effect the accuracy of variational approximation:

3.4.1.1 W: Like we discussed earlier, the variational approximation performs poorly when non-diagonal elements are present in the covariance matrix. The only way a distribution may end up having non-diagonal elements is if WW^T has non-diagonal elements. So, if W is such that WW^T, the variational approximation will be poor.

3.4.1.2 Ψ: Psi is the independent noise present at each dimension. The value of its elements, if sufficiently small, makes the distribution hard to estimate by the variational approximation, especially when WW^T is a singular matrix.

3.4.2 Settings of these variables which result in the highest KL divergence between approximate and true posteriors:

W: set W such that WW^T is singular:

```
W = np.array([[1, 1, 0],[1, 1, 0]])
```

Ψ: set it to a small value (as small as possible without failing the np.linalg.inv method for exact inference)

```
Psi = np.diag([1e-12, 1e-12, 1e-12])
```

This config results in a KL divergence of order 10^{12}, while the KL of 3.1 and 3.2 is around ~4

Experiment result with the above config:

Exact: **mean** = [0.5,0.5]

covariance = $\begin{bmatrix} 0.5 & -0.5 \\ -0.5 & 0.5 \end{bmatrix}$

BBSVI: **mean** = [0.4899635, 0.48227537]

covariance = $\begin{bmatrix} 0.24556576 & 0 \\ 0 & 0.24640145 \end{bmatrix}$

SVI_OBJ of last iteration: 1.01903419829e+12

Problem 04

4.1

Average marginal log likelihood for *training* data on *Exact fit*: 3023.59172638

Average marginal log likelihood for *test* data on *Exact fit*: 2936.61080806

Average marginal log likelihood for *training data* on *BBSVL fit*: 2993.00439427

Average marginal log likelihood for *test data* on *BBSVL fit*: 2918.09408461

4.2 Top 10 words for each K: exact learning

K	word	weight
0	movement	0.009894
0	subject	0.008513
0	human	0.008324
0	motor	0.00783
0	stimulus	0.006706
0	location	0.006562
0	motion	0.00642
0	eye	0.00606
0	orientation	0.005921
0	hand	0.00575
1	recurrent	0.006899
1	synaptic	0.00658
1	context	0.006575
1	sequence	0.005219
1	trajectory	0.005086
1	hmm	0.005069
1	firing	0.004961
1	movement	0.004905
1	forward	0.004546
1	motor	0.004484
2	motion	0.011507

2	stimulus	0.008203
2	spatial	0.007344
2	orientation	0.006821
2	cortex	0.006449
2	frequency	0.005785
2	receptive	0.005762
2	eye	0.005593
2	location	0.005283
2	pixel	0.005078
3	chip	0.012977
3	analog	0.009885
3	bit	0.00632
3	parallel	0.00629
3	activation	0.005828
3	implementation	0.005109
3	vlsi	0.004761
3	block	0.004651
3	recurrent	0.004485
3	propagation	0.004361
4	chip	0.009314
4	frequency	0.008983
4	analog	0.008288
4	motion	0.008009
4	synaptic	0.007932
4	spike	0.007765
4	voltage	0.007378
4	firing	0.007329
4	potential	0.005866
4	stimulus	0.005833
5	chip	0.009663
5	movement	0.008659
5	motor	0.008488
5	trajectory	0.007282
5	motion	0.006545
5	analog	0.005937
5	forward	0.005891
5	inverse	0.005843
5	arm	0.005446
5	hand	0.004667
6	table	0.005169
6	student	0.005076
6	validation	0.004125

6	teacher	0.003625
6	epoch	0.003616
6	cross	0.003207
6	synaptic	0.003191
6	rbf	0.003106
6	descent	0.003072
6	spike	0.002987
7	hmm	0.009678
7	context	0.008934
7	speaker	0.007786
7	chip	0.00728
7	analog	0.007079
7	mlp	0.007005
7	frequency	0.005812
7	acoustic	0.005434
7	filter	0.004678
7	classes	0.004436
8	theorem	0.00948
8	dimension	0.007384
8	proof	0.005965
8	polynomial	0.005848
8	distance	0.004788
8	perceptron	0.004223
8	kernel	0.00378
8	margin	0.00336
8	lemma	0.003275
8	tangent	0.003221
9	policy	0.017494
9	reinforcement	0.010762
9	decision	0.009037
9	reward	0.007368
9	controller	0.005881
9	robot	0.005872
9	markov	0.005726
9	goal	0.005548
9	sutton	0.005456
9	iteration	0.00533

4.3 Top 10 words for each K: BBSV learning

K	word	weight
0	attractor	0.001601
0	channel	0.001261

0	cross	0.001131
0	part	0.001055
0	goal	0.001039
0	inhibitory	0.001008
0	character	0.001006
0	separation	0.000987
0	ann	0.000956
1	table	0.001299
1	finite	0.00124
1	correlation	0.001233
1	letter	0.001188
1	string	0.001105
1	inference	0.001098
1	regression	0.001082
1	wavelet	0.001068
1	search	0.001043
1	previous	0.001018
2	tree	0.001226
2	nearest	0.001215
2	batch	0.001193
2	spike	0.001062
2	likelihood	0.000993
2	annealing	0.000969
2	adaptive	0.000963
2	analog	0.00096
2	hmm	0.00093
2	neighbor	0.000911
3	controller	0.001462
3	tree	0.001437
3	length	0.001332
3	kernel	0.001305
3	phase	0.001248
3	language	0.00113
3	positive	0.001111
3	tangent	0.001094
3	node	0.001075
3	cost	0.001043
4	filter	0.001602
4	cortex	0.001527
4	nonlinear	0.001434
4	belief	0.00136
4	expert	0.001323
4	chip	0.001298

4	principle	0.001182
4	potential	0.001109
4	head	0.001063
4	sensor	0.001008
5	filter	0.001851
5	iii	0.001236
5	synapses	0.001214
5	item	0.001084
5	scale	0.001044
5	natural	0.001034
5	complexity	0.001027
5	propagation	0.001027
5	spikes	0.001023
5	polynomial	0.000998
6	spike	0.00132
6	ieee	0.001073
6	channel	0.001066
6	sutton	0.001052
6	samples	0.001007
6	transistor	0.000993
6	ensemble	0.000965
6	character	0.000955
6	binary	0.000952
6	robot	0.000943
7	rbf	0.001693
7	stimulus	0.001508
7	potential	0.001482
7	hand	0.001269
7	center	0.001255
7	belief	0.001215
7	product	0.001164
7	nonlinear	0.001144
7	range	0.001098
7	feedback	0.001046
8	light	0.00142
8	basis	0.00139
8	reward	0.001292
8	mechanism	0.001054
8	strategy	0.001019
8	selection	0.001002
8	functional	0.000918
8	gain	0.000913
8	stimulus	0.000868

8	nonlinear	0.000816
9	sound	0.001389
9	rbf	0.001371
9	membrane	0.001337
9	trial	0.001291
9	cost	0.001189
9	attention	0.001152
9	auditory	0.001095
9	optimization	0.001077
9	reconstruction	0.001069
9	part	0.001053

4.4

Based on 4.1, the average marginal log likelihood is lower in BBSVL case for both train and test data, but not by much. This is an indicator of loss in expressive capabilities due to variational approximation.

As seen in 4.2, exact learning is very good at forming topic clusters. The top 10 words in each K do seem to be coming from a domain of ML/AI. For example, K=9 is a proper reinforcement learning word-set, K=8 has words belonging to computational theory, K=5 had words belonging to robotics and so on. In general, I observe all words in each K belong to the same topic, some topics have 1 word which may seem a bit off.

The quality of BBSV learning is not as good though. It does manage to come up with topics, but they either seem like combinations of topics rather than unique topics, or unique topics with 3-4 irrelevant words. For example, K=2 has general ML terms as top words, K=9 seems like a combination of robotics and RL, K=7 Seems mostly PGM, but has words like ‘hand’ and ‘stimulus’ which don’t seem to belong.