Detecting and Adapting to Irregular Distribution Shifts in Bayesian Online Learning

Aodong Li[1] Alex Boyd[2] Padhraic Smyth[1,2] Stephan Mandt[1,2]

[1]Department of Computer Science [2]Department of Statistics

University of California, Irvine

UCIrvine

NEURAL INFORMATION PROCESSING SYSTEMS
NeurIPS 2021

Motivation Examples

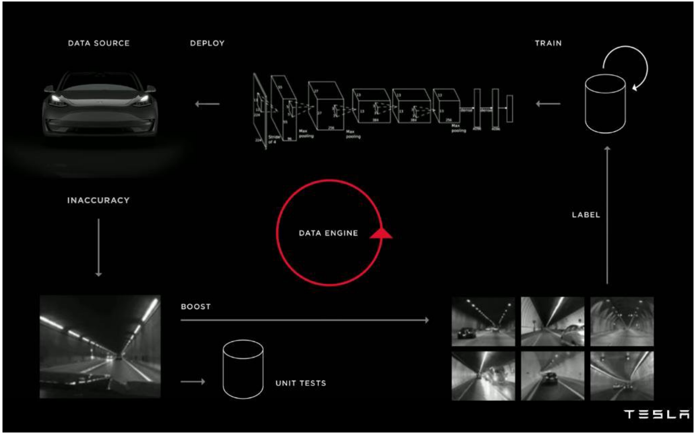
Learning in a sequential environment is important. Some practical examples include...

Motivation Examples

Learning in a sequential environment is important. Some practical examples include...

"work from home" before and during the pandemic.

Learning in a sequential environment is important. Some practical examples include...



[0] https://vimeo.com/274274744

Motivation Examples

Learning in a sequential environment is important. Some practical examples include...

Motivation Examples

Learning in a sequential environment is important. Some practical examples include...

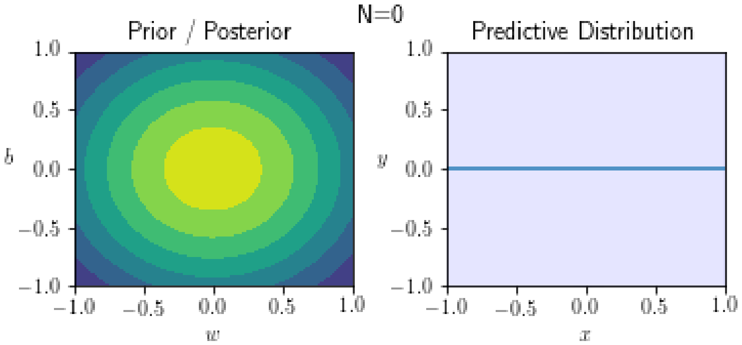
The environments are changing, which requires the model to update in an online fashion.

Bayesian Online Learning

Bayesian Online Learning

Repeatedly using Bayes' theorem naturally leads to an online learning framework

Repeatedly using Bayes' theorem naturally leads to an online learning framework



Bayesian Online Learning with Distribution Shift: the Problem

Bayesian online learning lacks efficiency in a changing environment.

Bayesian Online Learning with Distribution Shift: the Problem

Bayesian online learning lacks efficiency in a changing environment.
Reason: as the posterior shrinks when evidence accumulates, Bayesian online learning will get stuck with the first plausible solution.

Bayesian Online Learning with Distribution Shift: the Problem

Bayesian online learning lacks efficiency in a changing environment.
Reason: as the posterior shrinks when evidence accumulates, Bayesian online learning will get stuck with the first plausible solution.

Bayesian Online Learning with Distribution Shift: Solution

Introduce an additional step to allow for partial forgetting of the previous information.

Bayesian Online Learning with Distribution Shift: Solution

Introduce an additional step to allow for partial forgetting of the previous information.

Examples

Broaden the variance at every time step $\mathrm{Var}(\boldsymbol{z}) \quad {}^{1}\mathrm{Var}(\boldsymbol{z})$ where $\quad 2\ (0; 1)$ [Kulhavỳ and Zarrop, 1993, Kurle et al., 2020].

Bayesian Online Learning with Distribution Shift: Solution

Introduce an additional step to allow for partial forgetting of the previous information.

Examples

- Broaden the variance at every time step $\mathrm{Var}(\boldsymbol{z}) \quad {}^{1}\mathrm{Var}(\boldsymbol{z})$ where $\quad 2\ (0; 1)$ [Kulhavỳ and Zarrop, 1993, Kurle et al., 2020].
- Introduce additional noise [Welch et al., 1995] $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \quad {}_t$.

Bayesian Online Learning with Distribution Shift: Solution

Introduce an additional step to allow for partial forgetting of the previous information.

Examples

- Broaden the variance at every time step $\mathrm{Var}(z) \quad {}^{1}\mathrm{Var}(z)$ where $\ 2\ (0;1)$ [Kulhavy and Zarrop, 1993, Kurle et al., 2020].
- Introduce additional noise [Welch et al., 1995] $z_{t+1} = z_t + \ _t$.

However, the distribution shifts can vary at different rates, and the constant forgetting rate may not apply for all scenarios.

Model Assumption

To automatically determine when to adapt, we introduce a conditional prior for step t.

Model Assumption

To automatically determine when to adapt, we introduce a conditional prior for step t.

Model Assumption

To automatically determine when to adapt, we introduce a conditional prior for step t.

With a binary change variable $s_t \in \{0, 1\}$ and an inverse temperature $0 < \beta < 1$

$$p(z_t | s_t; \tau_t) = \begin{cases} \mathcal{N}(z_t; \mu_{t-1}, \sigma_{t-1}^2); & s_t = 0 \\ \mathcal{N}(z_t; \mu_{t-1}, \beta^{-1}\sigma_{t-1}^2); & s_t = 1 \end{cases}$$

where τ_t extracts the previous posterior's mean $\mu_{t-1}(q_{t-1})$ and variance $\sigma_{t-1}(q_{t-1})$.

Model Assumption

Our model's joint distribution factorizes as follows:

$$p(x_{1:T}; z_{1:T}; s_{1:T}) = \prod_{t=1}^{T}$$

Model Assumption

Our model's joint distribution factorizes as follows:

$$p(x_{1:T}; z_{1:T}; s_{1:T}) = \prod_{t=1}^{T} p(s_t)$$

Model Assumption

Our model's joint distribution factorizes as follows:

$$p(x_{1:T}, z_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t) p(z_t | s_t, \; {}_t)$$

Model Assumption

Our model's joint distribution factorizes as follows:

$$p(x_{1:T}, z_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t)p(z_t|s_t, {}_t)p(x_t|z_t)$$

Model Assumption

Our model's joint distribution factorizes as follows:

$$p(x_{1:T}, z_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t)\, p(z_t | s_t, \ _t)\, p(x_t | z_t)$$

$_t = F[p(z_{t-1} | x_{1:t-1}, s_{1:t-1})]$. Throughout our work, we use a speci c form

$$_t \ \{ \ _{t-1}, \ _{t-1} \} \ \{ \mathrm{Mean}, \mathrm{Var} \}[z_{t-1} | x_{1:t-1}, s_{1:t-1}]$$

Infer the distribution shift at step t

Simple in a tractable model! Similar to a likelihood-ratio test!

Simple in a tractable model! Similar to a likelihood-ratio test!

The posterior of s_t is again a Bernoulli distribution $p(s_t | s_{1:t-1}; x_{1:t}) = \text{Bern}(s_t; m)$

$$m = \log \frac{p(x_t | s_t = 1; s_{1:t-1}; x_{1:t-1})}{p(x_t | s_t = 0; s_{1:t-1}; x_{1:t-1})} + {}_0 \; ;$$

${}_0 = \log p(s_t = 1) - \log p(s_t = 0)$ are the log-odds of the prior $p(s_t)$.

Simple in a tractable model! Similar to a likelihood-ratio test!

The posterior of s_t is again a Bernoulli distribution $p(s_t | s_{1:t-1}, x_{1:t}) = \text{Bern}(s_t; m)$

$$m = \quad \log \frac{p(x_t | s_t = 1, s_{1:t-1}, x_{1:t-1})}{p(x_t | s_t = 0, s_{1:t-1}, x_{1:t-1})} + {}_0 \quad ;$$

${}_0 = \log p(s_t = 1) \quad \log p(s_t = 0)$ are the log-odds of the prior $p(s_t)$.

Same in an intractable model with variational inference!

Infer the distribution shift at step t

Simple in a tractable model! Similar to a likelihood-ratio test!

The posterior of s_t is again a Bernoulli distribution $p(s_t \mid s_{1:t\ \ 1}; x_{1:t}) = \ \mathrm{Bern}(s_t; m)$

$$m \ = \qquad \log \frac{p(x_t \mid s_t = 1; s_{1:t\ \ 1}; x_{1:t\ \ 1})}{p(x_t \mid s_t = 0; s_{1:t\ \ 1}; x_{1:t\ \ 1})} + \ {}_0 \quad ;$$

${}_0 = \log p(s_t \ = \ 1) \quad \log p(s_t \ = \ 0)$ are the log-odds of the prior $p(s_t)$.

Same in an intractable model with variational inference!

The variational posterior of s_t is also a Bernoulli distribution $\mathrm{Bern}(s_t; m)$

$$m = \qquad \log \frac{\exp L(q\ \ (z_t) \mid s_t \ = \ 1; s_{1:t\ \ 1})}{\underbrace{\exp L(q\ \ (z_t) \mid s_t \ = \ 0; s_{1:t\ \ 1})}_{p(x_t \mid s_t \ = \ 0; s_{1:t\ \ 1}; x_{1:t\ \ 1})}} + \ {}_0 \quad ;$$

Exponential Branching and Greedy Search

At time step t, the posterior branches into two configurations:

$$\begin{cases} s_t = 0: & p(z_t|s_t = 0, x_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 0|x_{1:t}, s_{1:t-1}) \\ s_t = 1: & p(z_t|s_t = 1, x_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 1|x_{1:t}, s_{1:t-1}) \end{cases}$$

Exponential Branching and Greedy Search

At time step t, the posterior branches into two configurations:

$$\begin{cases} s_t = 0: & p(z_t | s_t = 0, x_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 0 | x_{1:t}, s_{1:t-1}) \\ s_t = 1: & p(z_t | s_t = 1, x_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 1 | x_{1:t}, s_{1:t-1}) \end{cases}$$

Exponential branching prevents feasible computation.

At time step t, the posterior branches into two configurations:

$$\begin{cases} s_t = 0: & p(z_t | s_t = 0, x_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 0 | x_{1:t}, s_{1:t-1}) \\ s_t = 1: & p(z_t | s_t = 1, x_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 1 | x_{1:t}, s_{1:t-1}) \end{cases}$$

Greedy Search

Beam Search

Exact Beam Search for $s_{1:t}$

$$p(s_{1:t}|x_{1:t}) \propto p(s_t|x_{1:t}, s_{1:t-1})\, p(s_{1:t-1}|x_{1:t-1})$$

where $p(s_t|x_{1:t}, s_{1:t-1}) = \mathrm{Bern}(s_t; m)$ and

$$m = \quad \log \frac{p(x_t|s_t = 1, s_{1:t-1}, x_{1:t-1})}{p(x_t|s_t = 0, s_{1:t-1}, x_{1:t-1})} + {}_0$$

Beam Search

Exact Beam Search for $s_{1:t}$

$$p(s_{1:t}|x_{1:t}) \propto p(s_t|x_{1:t}; s_{1:t\ 1})\,p(s_{1:t\ 1}|x_{1:t\ 1})$$

where $p(s_t|x_{1:t}; s_{1:t\ 1}) = \mathrm{Bern}(s_t; m)$ and

$$m = \quad \log \frac{p(x_t|s_t = 1; s_{1:t\ 1}; x_{1:t\ 1})}{p(x_t|s_t = 0; s_{1:t\ 1}; x_{1:t\ 1})} + {}_0$$

Variational Beam Search for $s_{1:t}$

$$p(s_{1:t}|x_{1:t}) \propto q\ (s_t|s_{1:t\ 1})\,p(s_{1:t\ 1}|x_{1:t\ 1})$$

where $q\ (s_t|s_{1:t\ 1}) = \mathrm{Bern}(s_t; m)$ and

$$m = \quad \log \frac{\exp L(q\ (z_t)|s_t = 1; s_{1:t\ 1})}{\underbrace{\exp L(q\ (z_t)|s_t = 0; s_{1:t\ 1})}_{p(x_t|s_t = 0;\, s_{1:t\ 1};\, x_{1:t\ 1})}} + {}_0$$

Beam Search: Example

Beam search can correct the decisions in hindsight:

Experiments (1)

Detect the changes in word meanings using dynamic word embeddings[1].

an online version of word2vec[2]

---

[1]Bamler and Mandt, Dynamic Word Embeddings, ICML 2017

[2]Mikolov et al., Distributed Representations of Words and Phrases and their Compositionality, NIPS 2013

Experiments (3)

$$p(z_t | s_t;\ {}_t) = \begin{cases} \mathcal{N}(z_t;\ {}_{t\ 1};\ {}^{2}_{t\ 1}); & s_t = 0 \\ \mathcal{N}(z_t;\ {}_{t\ 1};\ {}^{1}\ {}^{2}_{t\ 1}); & s_t = 1 \end{cases}$$

Different temperature parameter gives rise to qualitatively different detected changes:

Experiments (3)

$$p(z_t | s_t; \;\; _t) = \begin{cases} N(z_t; \;\; _{t \;\; 1}; \;\; ^{2}_{t \;\; 1}); & s_t = 0 \\ N(z_t; \;\; _{t \;\; 1}; \;\; ^{1} \;\; ^{2}_{t \;\; 1}); & s_t = 1 \end{cases}$$

Different temperature parameter gives rise to qualitatively different detected changes:

Experiments (3)

$$p(z_t j s_t; \ _t) = \begin{cases} N(z_t; \ _t \ _1; \ ^2_t \ _1); & s_t = 0 \\ N(z_t; \ _t \ _1; \ ^1 \ ^2_t \ _1); & s_t = 1 \end{cases}$$

Different temperature parameter gives rise to qualitatively different detected changes:

Experiments (3)

$$p(z_t|s_t;\ _t) = \begin{cases} N(z_t;\ _{t\ 1};\ ^{2}_{t\ 1}); & s_t = 0 \\ N(z_t;\ _{t\ 1};\ \ ^{1}\ ^{2}_{t\ 1}); & s_t = 1 \end{cases}$$

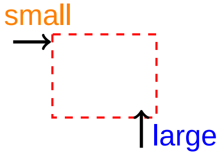
Different temperature parameter gives rise to qualitatively different detected changes:

small

Experiments (3)

$$p(z_t | s_t;\ {}_t) = \begin{cases} N(z_t;\ {}_t\ {}_1;\ {}^2_t\ {}_1); & s_t = 0 \\ N(z_t;\ {}_t\ {}_1;\ \ {}^1\ {}^2_t\ {}_1); & s_t = 1 \end{cases}$$

Different temperature parameter gives rise to qualitatively different detected changes:



Experiments (4)

Adapt to covariate shifts in supervised learning:

Experiments (5)

Table: Evaluation of Different Datasets

Models	CIFAR-10		SVHN		Malware	SensorDrift	Elec2	NBAPlayer	
	(Accuracy) ″					(MCAE 10 ²)#		(LogLike 10 ²) ″	
VBS (K=6)	**69.2**	**0.9**	**89.6**	**0.5**	**11.61**	**10.53**	7.28	**29.49**	**3.12**
VBS (K=3)	68.9	0.9	89.1	0.5	11.65	10.71	7.28	**29.22**	**2.63**
VBS (K=1)	68.2	0.8	88.9	0.5	11.65	10.86	**7.27**	**29.25**	**2.59**
BOCD (K=6)[J]	65.6	0.8	88.2	0.5	12.93	24.34	12.49	22.96	7.42
BOCD (K=3)[J]	67.3	0.8	88.8	0.5	12.74	24.31	12.49	20.93	7.83
BF[{]	**69.8**	**0.8**	**89.9**	**0.5**	11.71	11.40	13.37	24.17	2.29
VCL[y]	66.7	0.8	88.7	0.5	13.27	24.90	16.59	3.48	25.53
LP[z]	62.6	1.0	82.8	0.9	13.27	24.90	16.59	3.48	25.53
IB[x]	63.7	0.5	85.5	0.7	16.6	27.71	12.48	-44.87	16.88
IB[x] (Bayes)	64.5	0.3	87.8	0.1	16.6	27.71	12.48	-44.87	16.88

proposed, [J] [Adams and MacKay, 2007], [{] [Kurle et al., 2020]
[y] [Nguyen et al., 2018], [z] [Smola et al., 2003], [x] Independent Batch

Experiments (5)

Table: Evaluation of Different Datasets

MODELS	CIFAR-10		SVHN		MALWARE	SENSORDRIFT	ELEC2	NBAPLAYER	
	(ACCURACY)″					(MCAE 10^{2})#		(LOGLIKE 10^{2})″	
VBS (K=6)	**69.2**	**0.9**	**89.6**	**0.5**	**11.61**	**10.53**	7.28	**29.49**	**3.12**
VBS (K=3)	68.9	0.9	89.1	0.5	11.65	10.71	7.28	**29.22**	**2.63**
VBS (K=1)	68.2	0.8	88.9	0.5	11.65	10.86	**7.27**	**29.25**	**2.59**
BOCD (K=6)[J]	65.6	0.8	88.2	0.5	12.93	24.34	12.49	22.96	7.42
BOCD (K=3)[J]	67.3	0.8	88.8	0.5	12.74	24.31	12.49	20.93	7.83
BF[{]	**69.8**	**0.8**	**89.9**	**0.5**	11.71	11.40	13.37	24.17	2.29
VCL[y]	66.7	0.8	88.7	0.5	13.27	24.90	16.59	3.48	25.53
LP[z]	62.6	1.0	82.8	0.9	13.27	24.90	16.59	3.48	25.53
IB[x]	63.7	0.5	85.5	0.7	16.6	27.71	12.48	-44.87	16.88
IB[x] (BAYES)	64.5	0.3	87.8	0.1	16.6	27.71	12.48	-44.87	16.88

PROPOSED, [J] [ADAMS AND MACKAY, 2007], [{] [KURLE ET AL., 2020]
[y] [NGUYEN ET AL., 2018], [z] [SMOLA ET AL., 2003], [x] INDEPENDENT BATCH

Experiments (5)

Table: Evaluation of Different Datasets

MODELS	CIFAR-10		SVHN		MALWARE	SENSORDRIFT	ELEC2	NBAPLAYER	
	(ACCURACY)″				(MCAE 10 2)#			(LOGLIKE 10 2)″	
VBS (K=6)	**69.2**	**0.9**	**89.6**	**0.5**	**11.61**	**10.53**	7.28	**29.49**	**3.12**
VBS (K=3)	68.9	0.9	89.1	0.5	11.65	10.71	7.28	**29.22**	**2.63**
VBS (K=1)	68.2	0.8	88.9	0.5	11.65	10.86	**7.27**	**29.25**	**2.59**
BOCD (K=6)[J]	65.6	0.8	88.2	0.5	12.93	24.34	12.49	22.96	7.42
BOCD (K=3)[J]	67.3	0.8	88.8	0.5	12.74	24.31	12.49	20.93	7.83
BF[{]	**69.8**	**0.8**	**89.9**	**0.5**	11.71	11.40	13.37	24.17	2.29
VCL[y]	66.7	0.8	88.7	0.5	13.27	24.90	16.59	3.48	25.53
LP[z]	62.6	1.0	82.8	0.9	13.27	24.90	16.59	3.48	25.53
IB[x]	63.7	0.5	85.5	0.7	16.6	27.71	12.48	-44.87	16.88
IB[x] (BAYES)	64.5	0.3	87.8	0.1	16.6	27.71	12.48	-44.87	16.88

PROPOSED, [J] [ADAMS AND MACKAY, 2007], [{] [KURLE ET AL., 2020]
[y] [NGUYEN ET AL., 2018], [z] [SMOLA ET AL., 2003], [x] INDEPENDENT BATCH

Experiments (5)

Table: Evaluation of Different Datasets

MODELS	CIFAR-10 (ACCURACY) ″		SVHN		MALWARE	SENSORDRIFT (MCAE 10 ^{2})#	ELEC2	NBAPLAYER (LOGLIKE 10 ^{2}) ″	
VBS (K=6)	**69.2**	**0.9**	**89.6**	**0.5**	**11.61**	**10.53**	7.28	**29.49**	**3.12**
VBS (K=3)	68.9	0.9	89.1	0.5	11.65	10.71	7.28	**29.22**	**2.63**
VBS (K=1)	68.2	0.8	88.9	0.5	11.65	10.86	**7.27**	**29.25**	**2.59**
BOCD (K=6)[]]	65.6	0.8	88.2	0.5	12.93	24.34	12.49	22.96	7.42
BOCD (K=3)[]]	67.3	0.8	88.8	0.5	12.74	24.31	12.49	20.93	7.83
BF[{]	**69.8**	**0.8**	**89.9**	**0.5**	11.71	11.40	13.37	24.17	2.29
VCL[y]	66.7	0.8	88.7	0.5	13.27	24.90	16.59	3.48	25.53
LP[z]	62.6	1.0	82.8	0.9	13.27	24.90	16.59	3.48	25.53
IB[x]	63.7	0.5	85.5	0.7	16.6	27.71	12.48	-44.87	16.88
IB[x] (BAYES)	64.5	0.3	87.8	0.1	16.6	27.71	12.48	-44.87	16.88

PROPOSED, []] [ADAMS AND MACKAY, 2007], [{] [KURLE ET AL., 2020]
[y] [NGUYEN ET AL., 2018], [z] [SMOLA ET AL., 2003], [x] INDEPENDENT BATCH

Conclusion

We introduced a Bayesian inference algorithm for online learning in a non-stationary environment with irregular shifts.

Conclusion

We introduced a Bayesian inference algorithm for online learning in a non-stationary environment with irregular shifts.

Our approach simultaneously detect and adapt to shifts.

Conclusion

- We introduced a Bayesian inference algorithm for online learning in a non-stationary environment with irregular shifts.
- Our approach simultaneously detect and adapt to shifts.
- We introduced two schemes – greedy search and beam search – that trade expressiveness off against computation.

Conclusion

- We introduced a Bayesian inference algorithm for online learning in a non-stationary environment with irregular shifts.
- Our approach simultaneously detect and adapt to shifts.
- We introduced two schemes – greedy search and beam search – that trade expressiveness off against computation.
- Experiments show that our approach achieves lower error in supervised learning and compressive, interpretable latent structure in unsupervised learning.

References

R Kulhavỳ and Martin B Zarrop. On a general concept of forgetting. *International Journal of Control*, 58(4):905–924, 1993.

Richard Kurle, Botond Cseke, Alexej Klushyn, Patrick van der Smagt, and Stephan Günnemann. Continual learning with bayesian neural networks for non-stationary data. In *International Conference on Learning Representations*, 2020.

Greg Welch, Gary Bishop, et al. An introduction to the kalman filter. 1995.