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

Motivation Examples

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

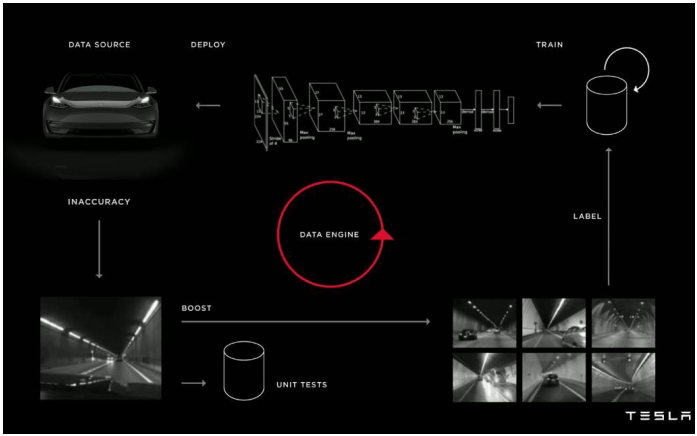
Motivation Examples

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

"work from home" before and during the pandemic.

Motivation Examples

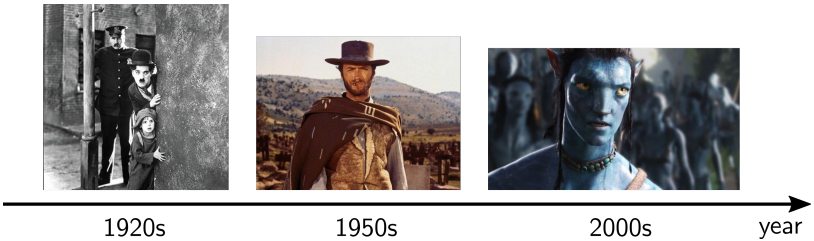
Learning in a sequential environment is important. Some practical examples include...



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

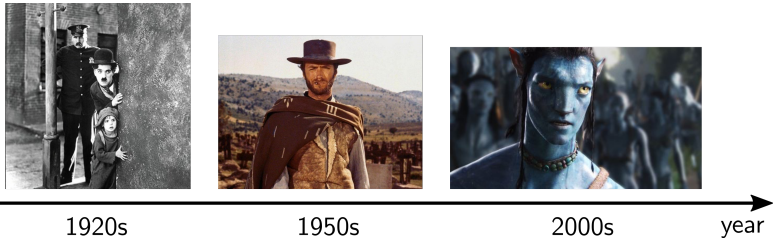
Motivation Examples

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.

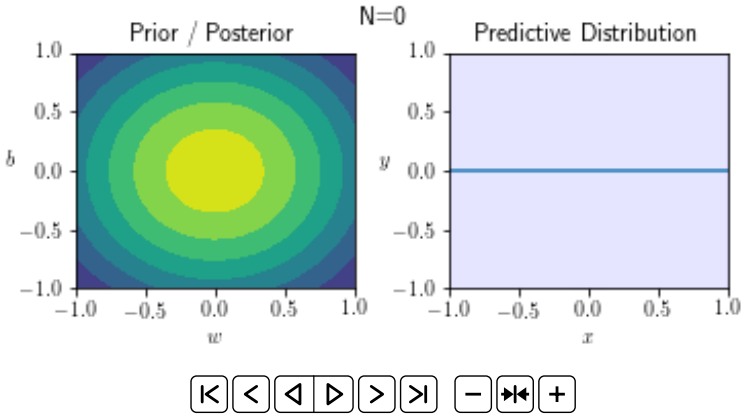
Bayesian Online Learning

Bayesian Online Learning

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

Bayesian Online Learning

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



Bayesian Online Learning with Distribution Shift: the Problem

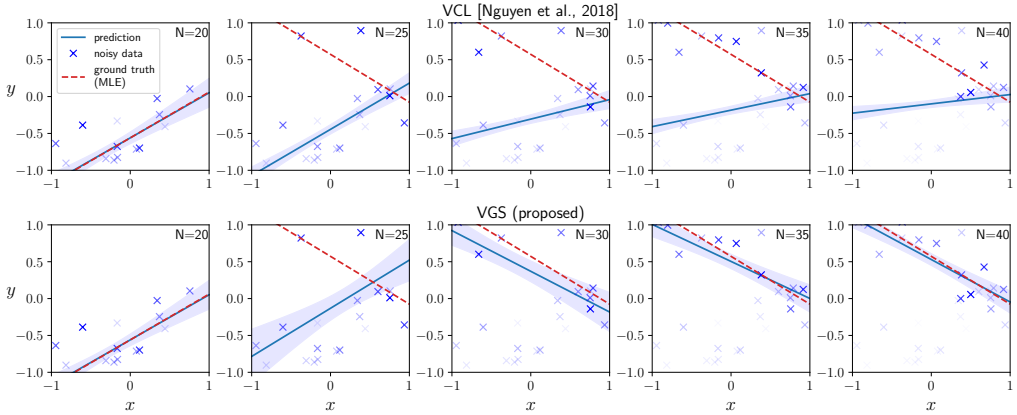
- Bayesian online learning lacks efficiency in a changing environment.

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 $\mathrm{Var}(\boldsymbol{z}) \leftarrow \beta^{-1}\mathrm{Var}(\boldsymbol{z})$ where $\beta \in (0, 1)$ [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 $\mathrm{Var}(\boldsymbol{z}) \leftarrow \beta^{-1}\mathrm{Var}(\boldsymbol{z})$ where $\beta \in (0, 1)$ [Kulhavỳ and Zarrop, 1993, Kurle et al., 2020].
- Introduce additional noise [Welch et al., 1995] $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \boldsymbol{\epsilon}_t$.

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 $\mathrm{Var}(\boldsymbol{z}) \leftarrow \beta^{-1}\mathrm{Var}(\boldsymbol{z})$ where $\beta \in (0, 1)$ [Kulhavỳ and Zarrop, 1993, Kurle et al., 2020].
- Introduce additional noise [Welch et al., 1995] $\boldsymbol{z}_{t+1} = \boldsymbol{z}_t + \boldsymbol{\epsilon}_t$.

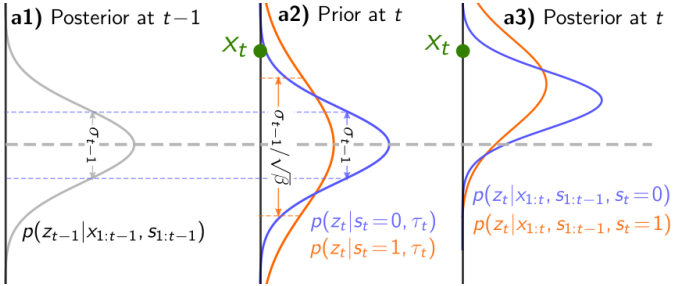
- *However, the distribution shifts can vary at different rates, and the constant forgetting rate may not apply for all scenarios.*

Model Assumption

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

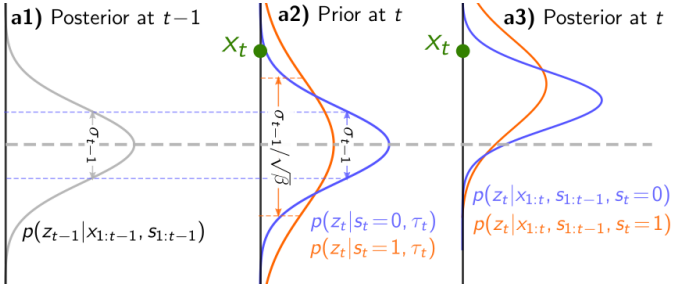
Model Assumption

- 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 $0 < \beta < 1$

$$p(\boldsymbol{z}_t|s_t; \boldsymbol{\tau}_t) = \begin{cases} \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_{t-1}, \sigma^2_{t-1}), & s_t = 0 \\ \mathcal{N}(\boldsymbol{z}_t; \boldsymbol{\mu}_{t-1}, \beta^{-1}\sigma^2_{t-1}), & s_t = 1 \end{cases}$$

where $\boldsymbol{\tau}_t$ extracts the previous posterior's mean $\boldsymbol{\mu}_{t-1}(q_{t-1})$ and variance $\sigma_{t-1}(q_{t-1})$.

Model Assumption

- Our model's joint distribution factorizes as follows:

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}, s_{1:T}) = \prod_{t=1}^{T}$$

Model Assumption

- Our model's joint distribution factorizes as follows:

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t)$$

Model Assumption

- Our model's joint distribution factorizes as follows:

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t) p(\boldsymbol{z}_t | s_t; \boldsymbol{\tau}_t)$$

Model Assumption

- Our model's joint distribution factorizes as follows:

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t) p(\boldsymbol{z}_t | s_t; \tau_t) p(\boldsymbol{x}_t | \boldsymbol{z}_t)$$

Model Assumption

- Our model's joint distribution factorizes as follows:

$$p(\boldsymbol{x}_{1:T}, \boldsymbol{z}_{1:T}, s_{1:T}) = \prod_{t=1}^{T} p(s_t) p(\boldsymbol{z}_t | s_t; \tau_t) p(\boldsymbol{x}_t | \boldsymbol{z}_t)$$

- $\tau_t = \mathcal{F}[p(\boldsymbol{z}_{t-1} | \boldsymbol{x}_{1:t-1}, s_{1:t-1})]$. Throughout our work, we use a specific form

$$\tau_t \equiv \{\boldsymbol{\mu}_{t-1}, \Sigma_{t-1}\} \equiv \{\text{Mean}, \text{Var}\}[\boldsymbol{z}_{t-1} | \boldsymbol{x}_{1:t-1}, s_{1:t-1}]$$

Infer the distribution shift at step t

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

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 $p(s_t|s_{1:t-1}, \boldsymbol{x}_{1:t}) = \mathrm{Bern}(s_t; m)$

$$m = \sigma\left(\log \frac{p(\boldsymbol{x}_t|s_t=1, s_{1:t-1}, \boldsymbol{x}_{1:t-1})}{p(\boldsymbol{x}_t|s_t=0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right),$$

$\xi_0 = \log p(s_t = 1) - \log p(s_t = 0)$ are the log-odds of the prior $p(s_t)$.

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 $p(s_t|s_{1:t-1}, \boldsymbol{x}_{1:t}) = \mathrm{Bern}(s_t; m)$

$$m = \sigma\left(\log \frac{p(\boldsymbol{x}_t|s_t=1, s_{1:t-1}, \boldsymbol{x}_{1:t-1})}{p(\boldsymbol{x}_t|s_t=0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right),$$

  $\xi_0 = \log p(s_t = 1) - \log p(s_t = 0)$ 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 $p(s_t|s_{1:t-1}, \boldsymbol{x}_{1:t}) = \mathrm{Bern}(s_t; m)$

$$m = \sigma\left(\log \frac{p(\boldsymbol{x}_t|s_t=1, s_{1:t-1}, \boldsymbol{x}_{1:t-1})}{p(\boldsymbol{x}_t|s_t=0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right),$$

$\xi_0 = \log p(s_t = 1) - \log p(s_t = 0)$ 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 $\mathrm{Bern}(s_t; m)$

$$m = \sigma\left(\log \underbrace{\frac{\exp \mathcal{L}(q^*(\boldsymbol{z}_t)|s_t = 1, s_{1:t-1})}{\exp \mathcal{L}(q^*(\boldsymbol{z}_t)|s_t = 0, s_{1:t-1})}}_{\approx p(\boldsymbol{x}_t|s_t = 0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right),$$

Exponential Branching and Greedy Search

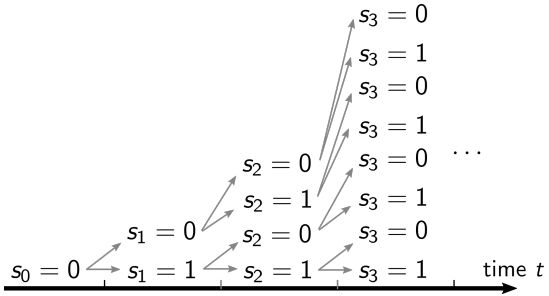
- At time step t, the posterior branches into two configurations:

$$\begin{cases} s_t = 0: & p(\boldsymbol{z}_t | s_t = 0, \boldsymbol{x}_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 0 | \boldsymbol{x}_{1:t}, s_{1:t-1}) \\ s_t = 1: & p(\boldsymbol{z}_t | s_t = 1, \boldsymbol{x}_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 1 | \boldsymbol{x}_{1:t}, s_{1:t-1}) \end{cases}$$

Exponential Branching and Greedy Search

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

$$\begin{cases} s_t = 0: & p(\boldsymbol{z}_t | s_t = 0, \boldsymbol{x}_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 0 | \boldsymbol{x}_{1:t}, s_{1:t-1}) \\ s_t = 1: & p(\boldsymbol{z}_t | s_t = 1, \boldsymbol{x}_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 1 | \boldsymbol{x}_{1:t}, s_{1:t-1}) \end{cases}$$



- Exponential branching prevents feasible computation.

Exponential Branching and Greedy Search

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

$$\begin{cases} s_t = 0: & p(\boldsymbol{z}_t | s_t = 0, \boldsymbol{x}_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 0 | \boldsymbol{x}_{1:t}, s_{1:t-1}) \\ s_t = 1: & p(\boldsymbol{z}_t | s_t = 1, \boldsymbol{x}_{1:t}, s_{1:t-1}) \text{ weighted by } p(s_t = 1 | \boldsymbol{x}_{1:t}, s_{1:t-1}) \end{cases}$$

Greedy Search

$$s_0 = 0 \rightarrow \begin{cases} s_1 = 0 \\ s_1 = 1 \rightarrow \end{cases} \begin{cases} s_2 = 0 \rightarrow \\ s_2 = 1 \end{cases} \begin{cases} s_3 = 0 \rightarrow \\ s_3 = 1 \end{cases} \begin{cases} s_4 = 0 \\ s_4 = 1 \rightarrow \end{cases} \begin{cases} s_5 = 0 \\ s_5 = 1 \rightarrow \cdots \end{cases}$$

Beam Search

Beam Search

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

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

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

$$m = \sigma\left(\log \frac{p(\boldsymbol{x}_t|s_t=1, s_{1:t-1}, \boldsymbol{x}_{1:t-1})}{p(\boldsymbol{x}_t|s_t=0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right)$$

Beam Search

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

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

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

$$m = \sigma\left(\log \frac{p(\boldsymbol{x}_t|s_t=1, s_{1:t-1}, \boldsymbol{x}_{1:t-1})}{p(\boldsymbol{x}_t|s_t=0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right)$$

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

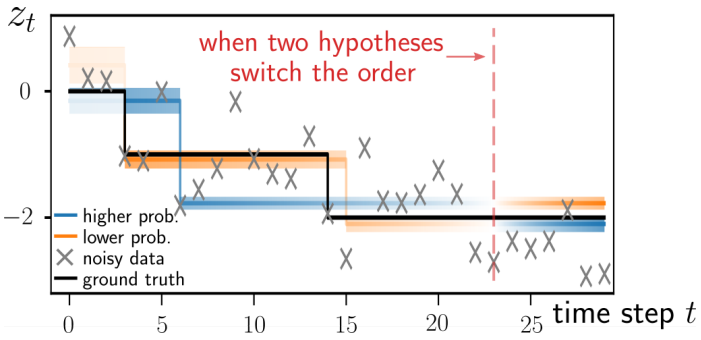
$$p(s_{1:t}|\boldsymbol{x}_{1:t}) \propto q^*(s_t|s_{1:t-1})p(s_{1:t-1}|\boldsymbol{x}_{1:t-1})$$

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

$$m = \sigma\left(\log \underbrace{\frac{\exp \mathcal{L}(q^*(\boldsymbol{z}_t)|s_t = 1, s_{1:t-1})}{\exp \mathcal{L}(q^*(\boldsymbol{z}_t)|s_t = 0, s_{1:t-1})}}_{\approx\, p(\boldsymbol{x}_t|s_t = 0, s_{1:t-1}, \boldsymbol{x}_{1:t-1})} + \xi_0\right)$$

Beam Search: Example

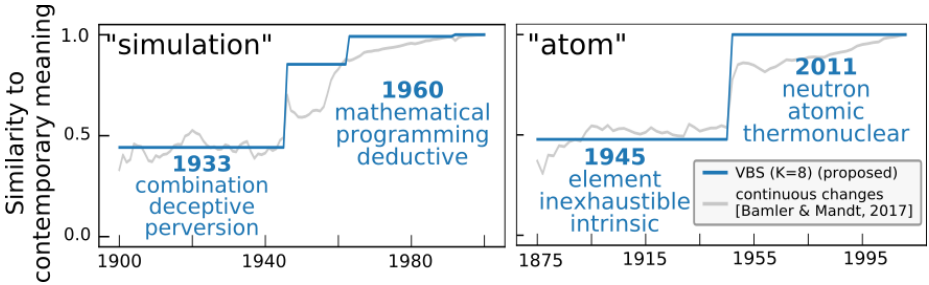
Beam search can correct the decisions in hindsight:



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)

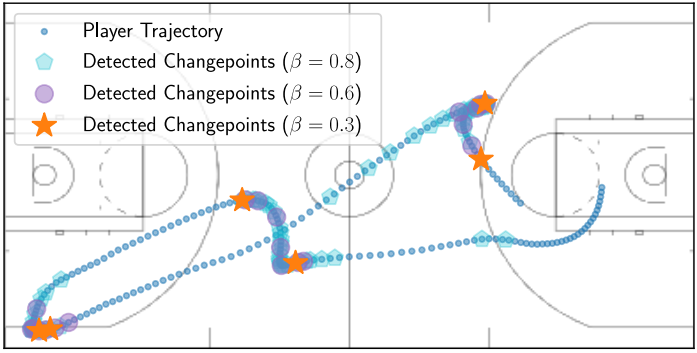
$$p(\boldsymbol{z}_t|s_t;\tau_t) = \begin{cases} \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\sigma^2_{t-1}), & s_t = 0 \\ \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\beta^{-1}\sigma^2_{t-1}), & s_t = 1 \end{cases}$$

Different temperature parameter gives rise to qualitatively different detected changes:

Experiments (3)

$$p(\boldsymbol{z}_t|s_t;\tau_t) = \begin{cases} \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\sigma^2_{t-1}), & s_t = 0 \\ \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\beta^{-1}\sigma^2_{t-1}), & s_t = 1 \end{cases}$$

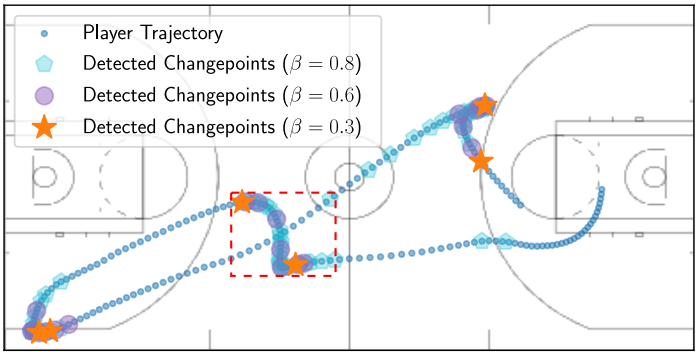
Different temperature parameter gives rise to qualitatively different detected changes:



Experiments (3)

$$p(\boldsymbol{z}_t|s_t;\tau_t) = \begin{cases} \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\sigma^2_{t-1}), & s_t = 0 \\ \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\beta^{-1}\sigma^2_{t-1}), & s_t = 1 \end{cases}$$

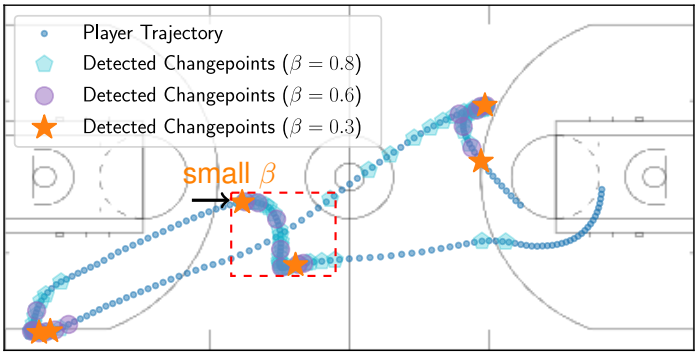
Different temperature parameter gives rise to qualitatively different detected changes:



Experiments (3)

$$p(\boldsymbol{z}_t|s_t;\tau_t) = \begin{cases} \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\sigma^2_{t-1}), & s_t = 0 \\ \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\beta^{-1}\sigma^2_{t-1}), & s_t = 1 \end{cases}$$

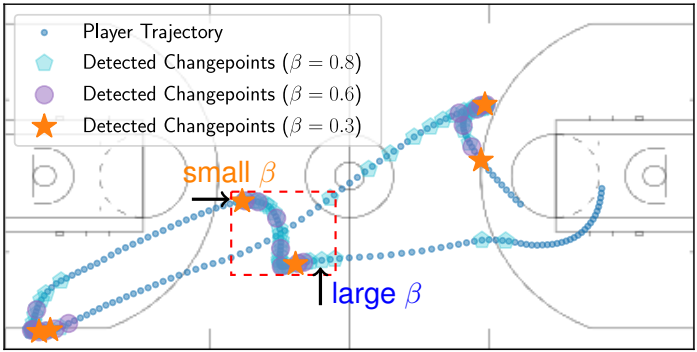
Different temperature parameter gives rise to qualitatively different detected changes:



Experiments (3)

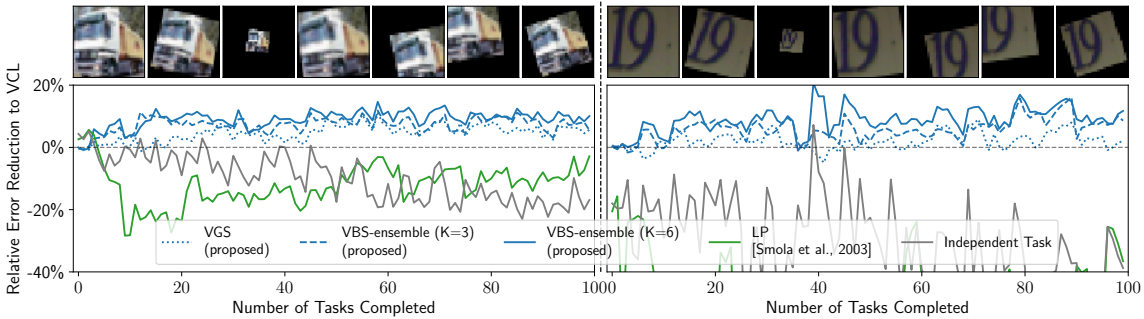
$$p(\boldsymbol{z}_t|s_t;\tau_t) = \begin{cases} \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\sigma^2_{t-1}), & s_t = 0 \\ \mathcal{N}(\boldsymbol{z}_t;\boldsymbol{\mu}_{t-1},\beta^{-1}\sigma^2_{t-1}), & s_t = 1 \end{cases}$$

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 (Accuracy)↑	SVHN (Accuracy)↑	Malware (MCAE 10^{-2})↓	SensorDrift (MCAE 10^{-2})↓	Elec2 (MCAE 10^{-2})↓	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†	66.7±0.8	88.7±0.5	13.27	24.90	16.59	3.48±25.53
LP‡	62.6±1.0	82.8±0.9	13.27	24.90	16.59	3.48±25.53
IB§	63.7±0.5	85.5±0.7	16.6	27.71	12.48	-44.87±16.88
IB§ (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]

† [Nguyen et al., 2018], ‡ [Smola et al., 2003], § Independent Batch

Experiments (5)

Table: Evaluation of Different Datasets

MODELS	CIFAR-10 (ACCURACY)↑	SVHN (ACCURACY)↑	MALWARE (MCAE 10^{-2})↓	SENSORDRIFT (MCAE 10^{-2})↓	ELEC2 (MCAE 10^{-2})↓	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†	66.7±0.8	88.7±0.5	13.27	24.90	16.59	3.48±25.53
LP‡	62.6±1.0	82.8±0.9	13.27	24.90	16.59	3.48±25.53
IB§	63.7±0.5	85.5±0.7	16.6	27.71	12.48	-44.87±16.88
IB§ (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]

† [NGUYEN ET AL., 2018], ‡ [SMOLA ET AL., 2003], § INDEPENDENT BATCH

Experiments (5)

Table: Evaluation of Different Datasets

MODELS	CIFAR-10 (ACCURACY)↑	SVHN (ACCURACY)↑	MALWARE (MCAE 10^{-2})↓	SENSORDRIFT (MCAE 10^{-2})↓	ELEC2 (MCAE 10^{-2})↓	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†	66.7±0.8	88.7±0.5	13.27	24.90	16.59	3.48±25.53
LP‡	62.6±1.0	82.8±0.9	13.27	24.90	16.59	3.48±25.53
IB§	63.7±0.5	85.5±0.7	16.6	27.71	12.48	-44.87±16.88
IB§ (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]

† [NGUYEN ET AL., 2018], ‡ [SMOLA ET AL., 2003], § INDEPENDENT BATCH

Experiments (5)

Table: Evaluation of Different Datasets

MODELS	CIFAR-10 (ACCURACY)↑	SVHN (ACCURACY)↑	MALWARE (MCAE 10^{-2})↓	SENSORDRIFT (MCAE 10^{-2})↓	ELEC2 (MCAE 10^{-2})↓	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†	66.7±0.8	88.7±0.5	13.27	24.90	16.59	3.48±25.53
LP‡	62.6±1.0	82.8±0.9	13.27	24.90	16.59	3.48±25.53
IB§	63.7±0.5	85.5±0.7	16.6	27.71	12.48	-44.87±16.88
IB§ (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]

† [NGUYEN ET AL., 2018], ‡ [SMOLA ET AL., 2003], § 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.