ThompsonBALD: Bayesian Batch Active Learning for Deep Learning via Thompson Sampling

Jaeik Jeon
DoAI
University College London
jaeikjeon@doai.ai

Brooks Paige
University College London
b.paige@ucl.ac.uk

Introduction

- Deep learning has been successfully applied to many pattern recognition tasks, but typically requires large labelled datasets and presents challenges in domains where acquiring labels is expensive.
- Probabilistic active learning methods aim to help by framing the labeling process as a decision problem, greedily selecting the most informative next data point to label.
- However, difficulties arise when considering a batch active learning setting: naive greedy approaches to batch construction can result in highly correlated queries.
- Several techniques for deep learning that are robust to batch selection have been proposed. [2] propose BatchBALD as an extension of BALD [1]. This takes information overlap of multiple points into account so that they are jointly informative.
- [3] construct a batch that best approximates the complete log posterior inspired by Bayesian coresets as sparse subset approximation

Contribution

- In this work, we introduce ThompsonBALD, a simple and surprisingly effective Bayesian batch active learning method, based on Thompson sampling of the mutual information between a data point and the model parameters.
- We demonstrate ThompsonBALD achieves performance comparable to other recent methods with significantly reduced computational time, and also compares favorably to other approaches which achieve batch diversity through injecting noise.

Materials and Methods

- Thompson sampling [4] is a general heuristic search algorithm that can trade off exploration vs exploitation by maximizing a reward conditional on a randomly sampled belief.
- We derive an algorithm for batch active learning by reformulating the BALD objective as an expected reward, and using Thompson sampling to select data points.
- The BALD acquisition function $a_{BALD}(\mathbf{x}, p(\boldsymbol{\omega}|D_0))$ is given and can be rearranged as

$$\begin{aligned}
\mathbb{I}[y, \boldsymbol{\omega}|\mathbf{x}, \mathcal{D}_0] &= \mathbb{H}[y|\mathbf{x}, \mathcal{D}_0] - \mathbb{E}_{\boldsymbol{\omega}|\mathcal{D}_0}[\mathbb{H}[y|\mathbf{x}, \mathcal{D}_0, \boldsymbol{\omega}]] && (1)\\
&= \mathbb{E}_{\boldsymbol{\omega}|\mathcal{D}_0}[\mathbb{H}[y|\mathbf{x}, \mathcal{D}_0] - \mathbb{H}[y|\mathbf{x}, \mathcal{D}_0, \boldsymbol{\omega}]] && (2)\\
&:= \mathbb{E}_{\boldsymbol{\omega}|\mathcal{D}_0}[\mathbb{E}[r^{\mathrm{BALD}}|\mathbf{x}, \mathcal{D}_0, \boldsymbol{\omega}]] && (3)
\end{aligned}$$

- With this perspective, the greedy action is taken by $\arg\max_{\mathbf{x}} \mathbb{E}_{\boldsymbol{\omega}|\mathcal{D}_0}[\mathbb{E}[r^{\mathrm{BALD}}|\mathbf{x}, \mathcal{D}_0, \boldsymbol{\omega}]]$. Therefore, the acquisition selection by BALD as in [1] enforces only exploitation by choosing the action that maximizes the expected reward.
- Thompson sampling for equation (3) is to use a single sample to approximate the expectation. However, utilising a single sample gives $\mathbb{I}[y, \boldsymbol{\omega}|\mathbf{x}, D_0] = 0$, since the marginal predictive entropy $\mathbb{H}[y|\mathbf{x}, \mathcal{D}_0]$ is also approximated by Monte Carlo.
- In this context, we will consider the marginal predictive entropy $\mathbb{H}[y|\mathbf{x}, \mathcal{D}_0]$ as an independent estimate to the conditional predictive entropy $\mathbb{H}[y|\mathbf{x}, D_0, \boldsymbol{\omega}]$. This is approximated with independent MC samples to the one with the Thompson samples used for computing $\mathbb{H}[y|\mathbf{x}, \mathcal{D}_0, \boldsymbol{\omega}]$.
- The conditional expectation of reward is now given as:

$$\mathbb{E}[r^{\mathrm{ThompsonBALD}}|\mathbf{x}, D_0, \boldsymbol{\omega}] := \hat{\mathbb{H}}[y|\mathbf{x}, D_0] - \mathbb{H}[y|\mathbf{x}, D_0, \boldsymbol{\omega}] \quad (4)$$

The corresponding pseudocode is shown in algorithm 1.

Algorithm 1 ThompsonBALD

Given $q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})$, $\hat{\mathbb{H}}[y|\mathbf{x}, \mathcal{D}_0]$, acquisition batch size B, $\mathcal{D}^{\mathrm{Batch}} = \emptyset$
for b=1,2,...,B **do**
  Sample $\hat{\boldsymbol{\omega}} \sim q^*_{\boldsymbol{\theta}}(\boldsymbol{\omega})$
  $\mathbf{x}_b^{\mathrm{Batch}} = \underset{\mathbf{x} \in \mathcal{D}^{\mathrm{Pool}} \setminus \mathcal{D}^{\mathrm{Batch}}}{\arg\max} \hat{\mathbb{H}}[y|\mathbf{x}, \mathcal{D}_0] - \mathbb{H}[y|\mathbf{x}, \mathcal{D}_0, \hat{\boldsymbol{\omega}}]$
  $\mathcal{D}^{\mathrm{Batch}} = \mathcal{D}^{\mathrm{Batch}} \cup \{\mathbf{x}_b^{\mathrm{Batch}}\}$
end for

- Thompson sampling with this expected reward chooses an action $\mathbf{x}$ with its probability of maximizing this disagreement.
- ThompsonBALD enforces both exploitation and exploration by taking variance into account produced by a single Monte Carlo sample from the posterior.

Results

- Our experiment follows the experimental setup in [2] and [3]. The experiment is done with repeated (Fashion) MNIST with added isotropic Gaussian noise with standard deviation 0.1.
- There are four baselines: Random, BALD, BatchBALD and ACS-FW. Only BatchBALD acquires 5 points in each active learning iteration in order to calculate the exact BatchBALD score. 10 random projections are used for ACS-FW.
- Every algorithm is evaluated on both MC dropout and neural linear uncertainty. All hyperparameters and the network structures are same with [2, 3].
- All models are trained with an initial training set of 50 data points, each time acquiring 10 and 20 points.
- ThompsonBALD can perform as well as or better than exact BatchBALD and ACS-FW; see figure 1 and 2.
- ThomsponBALD is significantly faster than other algorithms, as shown in table 1.
- BALD is highly sensitive to the types of uncertainty, and ThompsonBALD outperforms others on both uncertainties.
- ACS-FW does not actually produce the desired number of items per batch during each iteration. The curve in both figures are interpolated to the desired number.

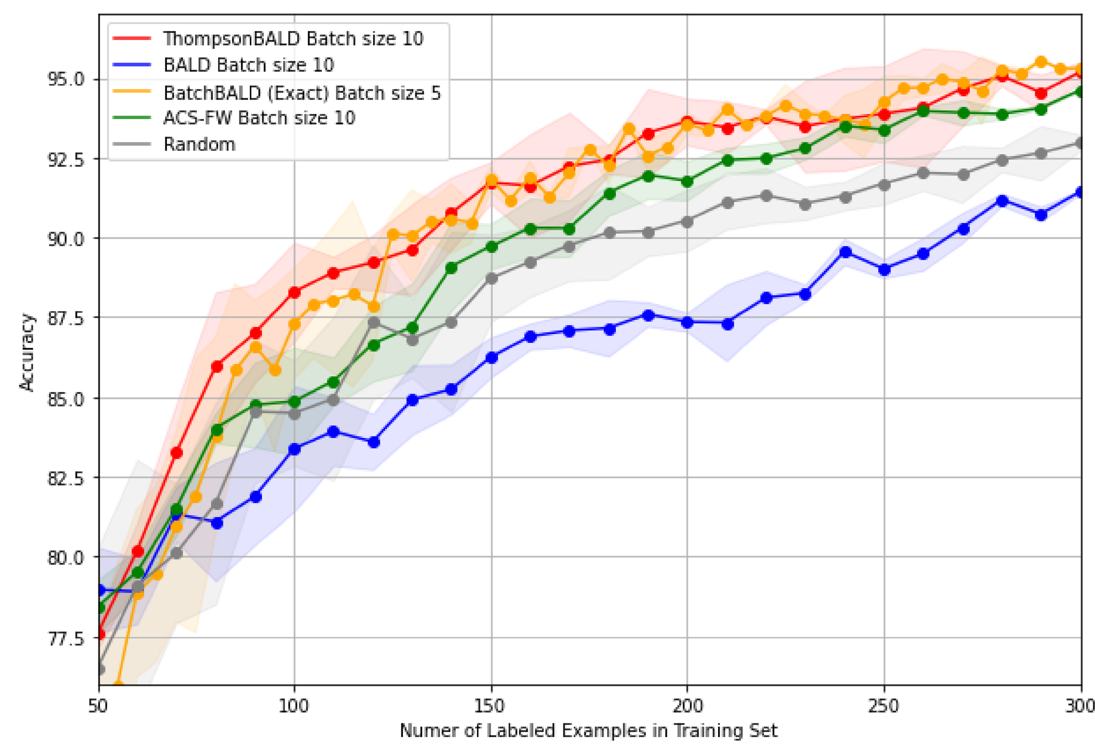

Figure 1: **Repeated MNIST test accuracy as a function of number of acquired points from the pool set with MC dropout uncertainty.**

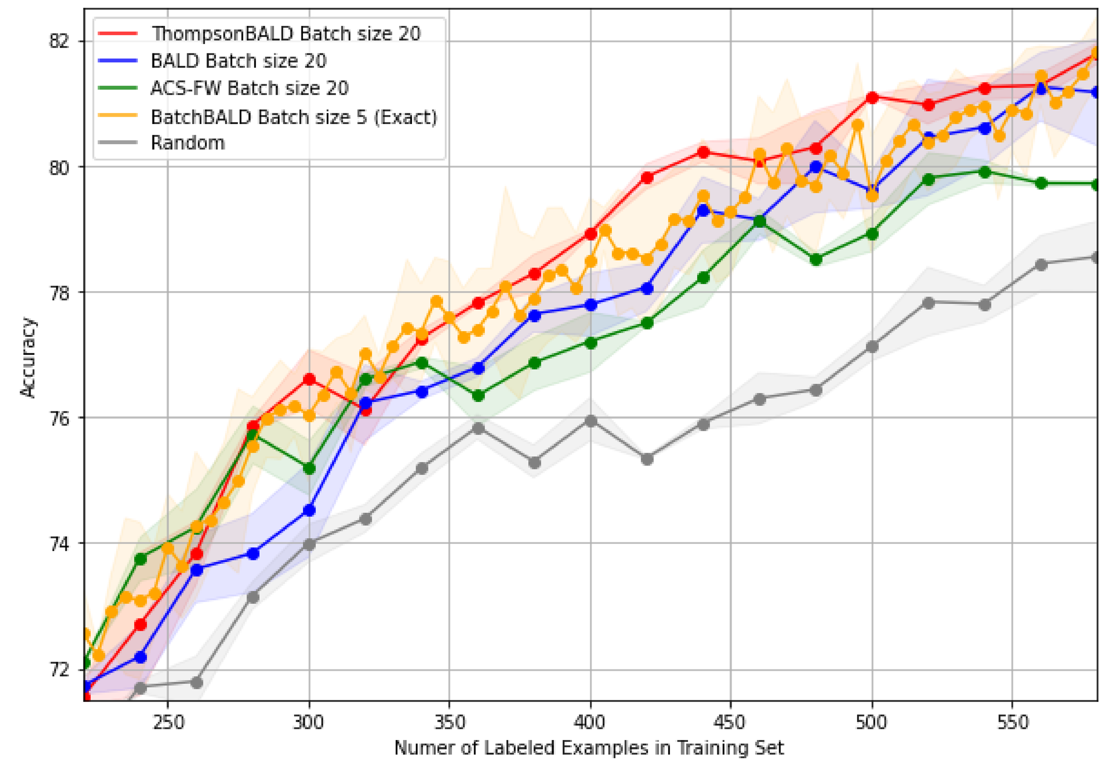

Figure 2: **Repeated Fashion MNIST test accuracy as a function of number of acquired points from the pool set with neural linear uncertainty.**

	BALD	BatchBALD	ACS-FW	ThompsonBALD
Elapsed time	2.4 ± 0.3	75.7 ± 0.3	2.5 ± 0.2	$\mathbf{1.2 \pm 0.03}$

Table 1: **Average elapsed time on each algorithm per acquisition iteration with one standard deviation.**

Conclusions

- We have introduced the simple and effective Bayesian batch active learning algorithm, ThompsonBALD.
- ThompsonBALD reduces the required computational time compared to BatchBALD and shows equivalent or improved performance over BALD, BatchBALD and ACS-FW.
- We expect ThompsonBALD can be a strong baseline as it is significantly fast and straightforward to implement.

References

[1] Gal, Y., Islam, R., and Ghahramani, Z. Deep bayesian active learning with image data. In *International Conference on Machine Learning* (2017), pp. 1183–1192.

[2] Kirsch, A., van Amersfoort, J., and Gal, Y. Batchbald: Efficient and diverse batch acquisition for deep bayesian active learning. In *Advances in Neural Information Processing Systems* (2019), pp. 7026–7037.

[3] Pinsler, R., Gordon, J., Nalisnick, E., and Hernández-Lobato, J. M. Bayesian batch active learning as sparse subset approximation. In *Advances in Neural Information Processing Systems* (2019), pp. 6359–6370.

[4] Thompson, W. R. On the likelihood that one unknown probability exceeds another in view of the evidence of two samples. *Biometrika 25*, 3/4 (1933), 285–294.