Scalable Batch-Mode Deep Bayesian Active Learning via Equivalence Class Annealing

Renyu Zhang, Aly A. Khan, Robert L. Grossman, Yuxin Chen

THE UNIVERSITY OF CHICAGO

Summary

- **BALanCe**: an efficient deep Bayesian active learning framework motivated by a decision-theoretic selection criterion
- **Batch-BALanCe:** batch-mode version

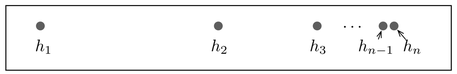
Motivation

Numerical example

The Most Information Selection criterion uses mutual information between the predicted label and model parameters as the acquisition function:

$$\Delta_{\mathrm{BALD}}(x \mid \mathscr{D}_{\mathrm{train}}) \triangleq \mathbb{I}(y; \omega \mid x, \mathscr{D}_{\mathrm{train}})$$

MIS/BALD can be ineffective

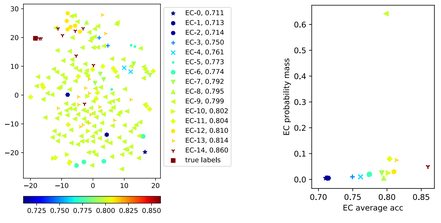


The hypothesis class $\mathscr{H} = \{h_1, \ldots, h_n\}$ is structured such that

$$d_{\mathrm{H}}(h_i, h_j) = \begin{cases} 2^{1-i} - 2^{1-j} & \text{if } i < j, \\ 2^{1-j} - 2^{1-i} & \text{o.w.} \end{cases}$$

- MIS/BALD on average require $O(\log n)$.
- A "smarter" policy could query examples to sequentially check the consistency of $h_1, h_2, \ldots, h_n$ util all remaining hypotheses are within distance σ. It requires $\log(1/\sigma)$.

Empirical validation



Samples from posterior BNN via MC dropout; embedding is generated by applying t-SNE on the disagreement between hypotheses;
Colorbar indicates the (approximate) test accuracy of the sampled neural networks on the MNIST dataset.

Problem statement

Assume

- labeling each query x incurs a unit cost.
- Labeled dataset $\mathscr{D}_{\mathrm{train}}$
- Unlabelled dataset $\mathscr{D}_{\mathrm{pool}}$ drawn i.i.d. from some underlying data distribution
- A set of hypotheses $\mathscr{H} = \{h_1, \ldots, h_n\}$;
- In this work, we consider BNN hypothesis class with parameters $\omega \sim \mathrm{p}(\omega \mid \mathscr{D}_{\mathrm{train}})$.

Want

an active learning algorithm that allows us to find a hypotheses with a target error rate $\sigma \in [0,1]$ while minimizing the total query cost.

Acquisition function

Threshold adaptively annealed based on validation prediction

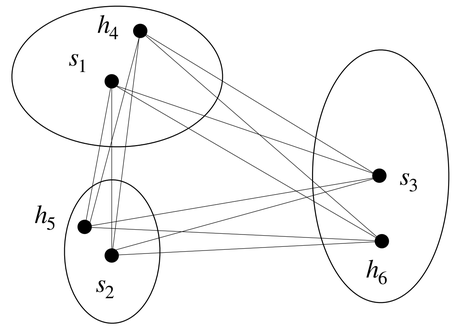
$$\Delta_{\mathrm{BALanCe}}(x_{1:b} \mid \mathscr{D}_{\mathrm{train}}) \approx \sum_{\hat{y}_{1:b}} \left(\sum_{k=1}^{K} \frac{p(\hat{y}_{1:b} \mid \hat{\omega}_k) + p(\hat{y}_{1:b} \mid \hat{\omega}'_k)}{2K} \right) \cdot \left[\sum_{k=1}^{K} \frac{1_{d_{\mathrm{H}}(\hat{\omega}_k, \hat{\omega}'_k) > \tau} \cdot \left(1 - \lambda_{\hat{\omega}_k \hat{y}_{1:b}} \lambda_{\hat{\omega}'_k \hat{y}_{1:b}}\right)}{K} \right]$$

New metric for clustering

$$\mathbb{I}_{\Delta_{\mathrm{BALanCe}}}(x_1, x_2) = \Delta_{\mathrm{BALanCe}}(x_1) + \Delta_{\mathrm{BALanCe}}(x_2) - \Delta_{\mathrm{BALanCe}}(\{x_1, x_2\})$$

Equivalence class

Definition 1.
Let $(\mathscr{H}, d)$ be a metric space where $\mathscr{H}$ is a hypothesis class and d is a metric. For a given set $\mathscr{V} \subseteq \mathscr{H}$ and centers $\mathscr{S} = \{s_1, \ldots, s_k\} \subseteq \mathscr{V}$ of size k, let $r^{\mathscr{S}} : \mathscr{V} \to [k]$ be a partition function over $\mathscr{V}$, and $\mathscr{D}_i := \{h \in \mathscr{V} \mid r^{\mathscr{S}}(h) = i\}$, such that $\forall i, j \in [k], r^{\mathscr{S}}(s_i) = i$ and $\forall h \in \mathscr{D}_i, d(h, s_i) \leq d(h, s_j)$. Each $\mathscr{D}_i \subseteq \mathscr{V}$ is called an equivalence class induced by $s_i \in \mathscr{S}$.



Two selection strategies

Algorithm I: Greedy selection

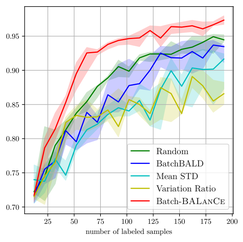
input: $\mathscr{D}_{\mathrm{pool}}$, $\bar{\mathscr{D}}_{\mathrm{pool}}$, a trained BNN, and threshold $\tau, \{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^{K}, B$
$\mathscr{A}_0 = \varnothing$
for $b \in [B]$ do
  for $x \in \mathscr{D}_{\mathrm{pool}} \backslash \mathscr{A}_{b-1}$ do
    $s_x \leftarrow \Delta_{BALanCe}(\mathscr{A}_{b-1} \cup \{x\})$
  $x_b \leftarrow \mathrm{argmax}_{x \in \mathscr{D} \backslash \mathscr{A}_{b-1}} s_x$
  $\mathscr{A}_b \leftarrow \mathscr{A}_{b-1} \cup \{x_b\}$
output: batch $\mathscr{A}_B$

For batch size < 40

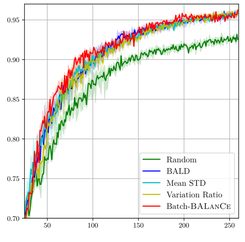
Algorithm II: BALanCe-Clustering

input: subset $\mathscr{C} \subseteq \mathscr{D}_{\mathrm{pool}}$, $\bar{\mathscr{D}}_{\mathrm{pool}}$, threshold $\tau, \{\hat{\omega}_k, \hat{\omega}'_k\}_{k=1}^{K}$, coldness parameter β, and B
sample initial centroids $\mathscr{O} = \{\mu_j\}_{j=1}^{B} \subseteq \mathscr{C}$ with $p(x) \sim \Delta_{\mathrm{BALanCe}}(x)^{\beta}$
while $\mathscr{O}$ not converged do
  for $x \in \mathscr{C}$ do
    $a_x \leftarrow \mathrm{argmax}_j \mathbb{I}_{\Delta_{\mathrm{BALanCe}}}(x, \mu_j)$
  $S_j \leftarrow \{x \in \mathscr{C} : a_x = j\}$
  for $j \in [B]$ do
    $\mu_j \leftarrow \mathrm{argmax}_{x_2 \in \mathscr{S}_j} \sum_{x_1 \in \mathscr{S}_j} \mathbb{I}_{\Delta_{\mathrm{BALanCe}}}(x_1, x_2)$
output: $\mathscr{S}_{1:B}, \mu_{1:B}$

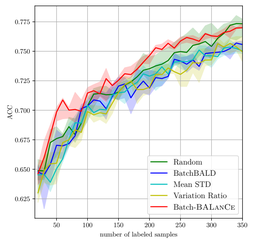
For batch size ≥ 40



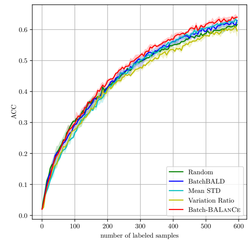
DRIFT; MC-dropout



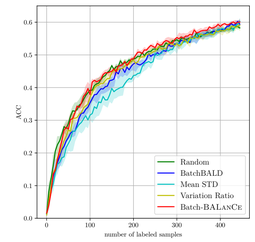
MNIST; MC-dropout



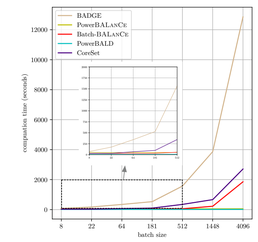
Fashion-MNIST; MC-dropout



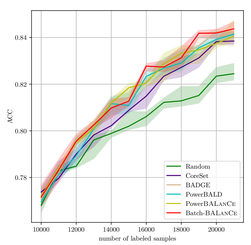
EMNIST-Balanced; MC-dropout



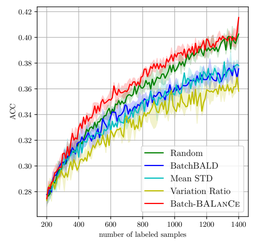
EMNIST-ByClass; MC-dropout



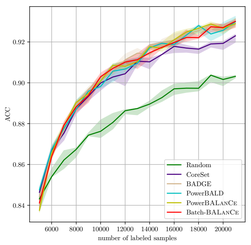
Computation time vs. batch size



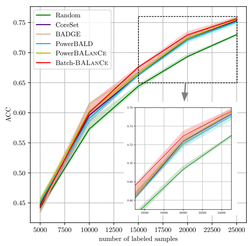
CIFAR-10; MC-dropout



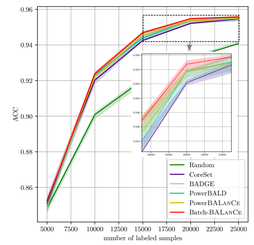
CINIC; MC-dropout



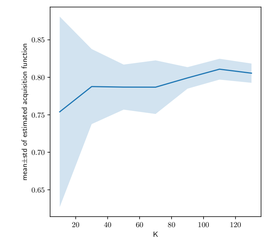
SVHN; MC-dropout



CIFAR-100; cSG-MCMC



CIFAR-10; cSG-MCMC



Acquisition function vs #MC samples