Bayesian Active Learning for Wearable and Mobile Health

Gautham Krishna Gudur†, Abhijith Ragav‡, Prahalathan Sundaramoorthy⋆, Venkatesh Umaashankar†
†Global AI Accelerator, Ericsson; ‡Solarillion Foundation; ⋆University of Southern California

Introduction

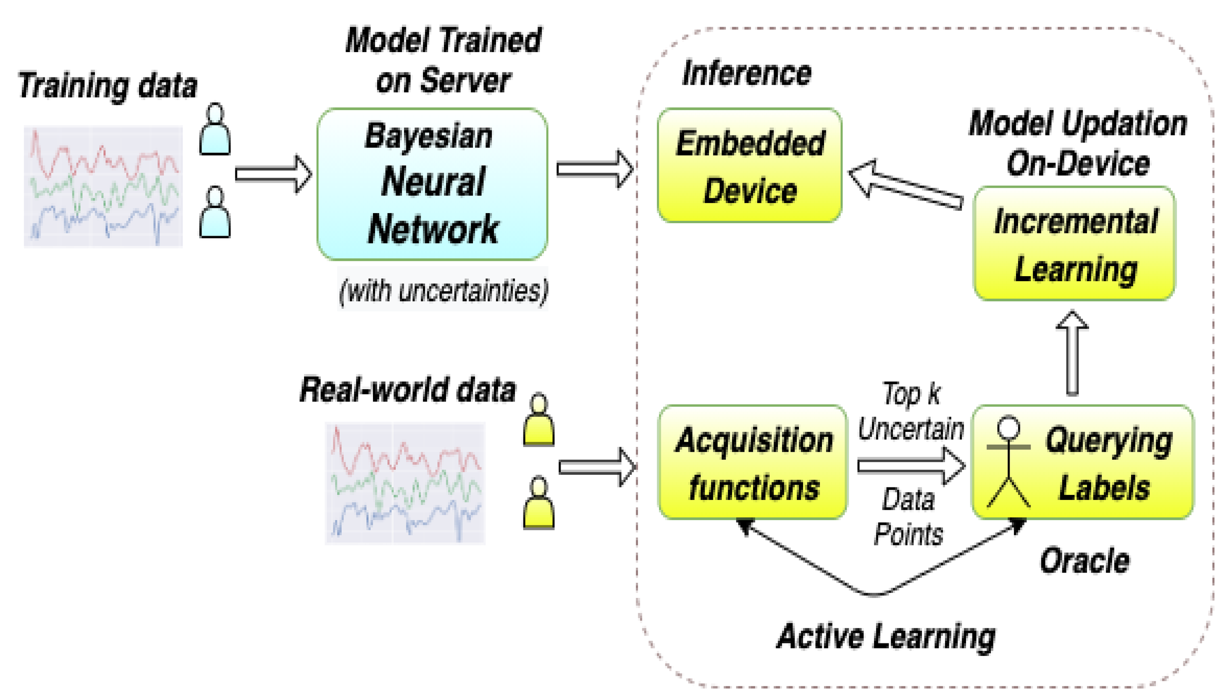
- Various mobile health applications require modeling of user behavior – **Human Activity Recognition (HAR)**, **Stress/Affect Detection**, **Fall Detection**, etc. [1, 2].
- ***Active Learning*** – the ability to learn from real-world unlabeled data by querying an oracle, is an unexplored area for such tasks.
- ***Bayesian Neural Networks (BNNs)*** – *Monte Carlo Dropout* with NNs to estimate predictive uncertainties [3].
- The predictive distribution for a new data point input x^* can be obtained by,

$$p(y^*|x^*, D_{train}) = \int p(y^*|x^*, \omega)p(\omega|D_{train})d\omega$$

where $p(\omega|D_{train}) = q^*_\theta(\omega)$, and $q^*_\theta(\omega)$ is the dropout distribution approximated using VI.

- *Dropout*, a light-weight operation enables easier and faster approximation of posterior uncertainties.
- Coupled with active learning *acquisition functions* for querying the most uncertain data points from the oracle.

Overall Block Diagram



Datasets

- *HHAR Smartwatch* – Wearable accelerometer data with 6 activities
- *SWELL-KW Stress/Affect Detection* – Heart Rate, Skin Conductance with 3 conditions
- *Notch Fall Detection* – Wrist-worn accelerometer data with falls and otherwise

Acquisition Functions

Given a classification model M, real-world pool data D_{pool}, and inputs $x \in D_{pool}$, an acquisition function $a(x, M)$ is a function of x that the active learning system uses to infer the next query point:

$$x^* = argmax_{x \in D_{pool}} a(x, M)$$

Acquisition functions are used in active learning scenarios for approximations in Bayesian CNNs, thereby arriving at the most efficient set of data points to query from D_{pool}.

Max Entropy: Pool points are chosen that maximize the predictive entropy.

$$\mathbb{H}[y|x, D_{train}] := -\sum_c p(y = c|x, D_{train}) \log p(y = c|x, D_{train})$$

Bayesian Active Learning by Disagreement (BALD): Pool points that maximize the mutual information between predictions and model posterior, that disagree the most about the outcome.

$$\mathbb{I}[y, \omega|x, D_{train}] = \mathbb{H}[y|x, D_{train}] - E_{p(\omega|D_{train})}[\mathbb{H}[y|x, \omega]]$$

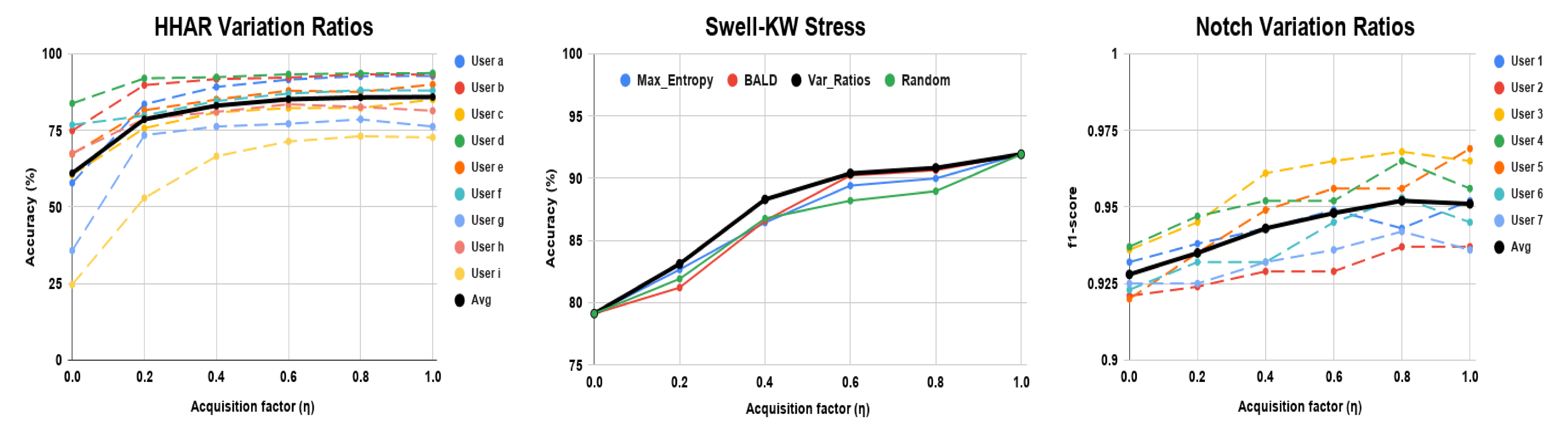
where $\mathbb{H}[y|x, \omega]$ is the entropy of y, given model weights ω.

Variation Ratios (VR): The LC (Least Confident) method for uncertainty based pool sampling.

$$variation - ratio[x] := 1 - \max_y p(y|x, D_{train})$$

Random Sampling: Select a point from a pool of data points uniformly at random.

Results



Model Description

- **HHAR Smartwatch and Notch Fall Detection Datasets:** We use the ***HARNet*** architecture [4] for both datasets, which explore intra-axial and inter-axial dependencies of CNNs.
- **SWELL-KW Dataset:** We use a four-layer Convolutional 1D network [1].
- Two fully-connected layers after the CNN models are used with a *MC-dropout* layer of probability 0.3 between them, making them ***Bayesian Convolutional Neural Networks (B-CNNs)***.
- To perform approximation inference in B-CNNs, dropout is performed at train and test-times using multiple stochastic forward passes (optimal dropout iterations – T=10).

On-Device Performance

- **Raspberry Pi 2** is used for evaluating our proposed active learning framework, due to similar HW/SW specifications as predominant contemporary mobile/wearable devices.
- Data split into D_{train}, D_{pool} and D_{test}. Leave-one-User-Out (LOOCV) and conventional data split at random is performed.
- **Acquisition adaptation factor** – $\eta \in$**[0, 1]**: The number of acquisition windows used for active learning from D_{pool}.
- *Variation Ratios* acquisition function yields best efficiencies across all datasets.
- Practical to threshold number of D_{pool} points collected in a single acquisition iteration. This can be quantified by either number of windows (w_a) or time taken (in seconds).

Table 1: Performance Metrics

Process	HHAR	Swell-KW	Notch
Baselines before AL	61%	79.12%	0.927
Inference time	14 ms	9 ms	11 ms
Discrete Wavelet Transform	0.5 ms	–	0.39 ms
Decimation	3.4 ms	–	–
Stochastic Forward Pass (T)	1.4 sec	0.5 sec	1 sec
Time taken per epoch	1.8 sec	0.6 sec	1.2 sec
Model size	315 kB	115 kB	180 kB

References

[1] Abhijith Ragav⋆ and Gautham Krishna Gudur⋆. Bayesian active learning for wearable stress and affect detection. In *NeurIPS Machine Learning for Mobile Health Workshop*, 2020.

[2] Gautham Krishna Gudur, Prahalathan Sundaramoorthy, and Venkatesh Umaashankar. Activeharnet: Towards on-device deep bayesian active learning for human activity recognition. In *The 3rd International Workshop on Deep Learning for Mobile Systems and Applications*, EMDL'19, 2019.

[3] Yarin Gal and Zoubin Ghahramani. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning*, ICML'16, 2016.

[4] Prahalathan Sundaramoorthy, Gautham Krishna Gudur, Manav Rajiv Moorthy, R. Nidhi Bhandari, and Vineeth Vijayaraghavan. Harnet: Towards on-device incremental learning using deep ensembles on constrained devices. In *Proceedings of the 2nd International Workshop on Embedded and Mobile Deep Learning*, EMDL'18, 2018.