**Online Linear Models for Edge Computing**

**Hadar Sivan¹, Moshe Gabel², Assaf Schuster¹**

¹Technion – Israel Institute of Technology, ²University of Toronto

---

**The Setting**

- Stream of (occasionally) labeled samples
- Computationally limited device
- Maintain an updated model from the stream
- Can offload some model computations to the cloud
- Data distribution may change over time

---

**Method and Tools**

- **Theorem 1.** Let $\Delta g$ be
  \[
  \Delta g := \sum_{i \in A} \nabla \ell_i(\beta^*_1) - \sum_{i \in R} \nabla \ell_i(\beta^*_1).
  \]
  Then:
  \[
  \|\beta^*_1 - \beta^*_2\| \leq 2\|r\|,
  \]
  where $r = \frac{1}{2} \left( \beta^*_1 - \frac{C_1}{C_2} \beta^*_1 + C_2 \Delta g \right)$.

---

**Our Solution**

**DRUiD – Drift detectR from boUnded Distance**

- Trades off a small number of batch model computations for better accuracy.
- Suitable for linear model with $L_2$-regularized convex differentiable loss.
- Supports both classification and regression tasks.

Monitors the **distance** between the **last batch-computed model** ($\beta_1^*$) and the **hypothetical model** that could be computed from the current position of the sliding window ($\beta_2^*$).

---

**DRUiD Algorithm**

For every new labeled sample:

1. Update the sliding window $W$
2. Update $\|r\|$ using Theorem 1
3. If $\|r\| > \text{Threshold}$, compute $\beta_1^*$ from the samples in $W$

Prediction for a new sample $x$:

1. Return the prediction of $\beta_1^*$ for $x$

---

**Results on Real data**

DRUiD achieves better tradeoff of accuracy vs number of computations than existing algorithms.

---

**Results on Ill-Conditioned Problems**

DRUiD’s batch mode performs well on ill-conditioned problems, while incremental algorithms perform poorly.