Edge-centric Efficient Regression Analytics

NATASCHA HARTH & CHRISTOS ANAGNOSTOPOULOS

SCHOOL OF COMPUTING SCIENCE
UNIVERSITY OF GLASGOW, SCOTLAND
Agenda

- Introduction
- Methodology
- Experiments
- Evaluation
- Results
Context

Sensing & Actuator Devices

Edge Device (ED)

Cloud Environments & Analytics

IoT Gateways (Edge Gateways (EG))

Introduction
Constraints at the Edge

1. Limited Bandwidth
2. Energy
3. Limited Computational Power
4. Storage Capacity

Idea: 
Observe Model Performance & Update the network Edge

- Exploit the limited computational power of Edge Devices
- Push Intelligence to the Edge:
  - inferential tasks, on-line statistical learning, classification, localized detection, ... are pushed at the Edge
Hypotheses & Actions

Given the constraints of an IoT network, let us hypothesise the following actions:

- **Action 1: Reduce** the communication overhead
  - **Hypothesis 1:** No raw data transfer is needed for inferential & regression analytics, i.e., **Learn More With Less!**

- **Action 2: Perform** real-time predictive & regression analytics for instant action & autonomous decision making
  - **Hypothesis 2:** use the limited computational power to infer and take decisions for regression models updates in an **On-Line Manner!**

- **Action 3: Provide** high quality predictive analytics tasks, e.g., prediction accuracy, model fitting
  - **Hypothesis 3:** decide which is the best diverse model to select based on given data statistics, i.e., **Be Intelligent On What You See!**
Challenges & Problem Definition

- Reduce **unnecessary** communication between/among devices and/or the Cloud
- **Problem 1:** Conditionally Model Forwarding Problem at the **Edge Device**

- Decide *which* model to select given all cached diverse models to maximize the predictive analytics accuracy
- **Problem 2:** Diverse Model Selection Problem at the **Edge Gateway**

- Decide *which* statistics to communicate *to support* the selection at the Gateway
- **Problem 3:** Time-optimized Data Selection Problem at the **Edge Gateway**

- Decide *when* to deliver/send updated models and *what* to send in light of maximizing the predictive analytics accuracy
- **Problem 4:** Time-optimized Model Delivery Scheduling at the **Edge Device**
Contribution

✓ Introduce an **communication efficient** scheme that transmits **only** regression model parameters & sufficient statistics in the Edge Network for cached model updates in Edge Gateways.

✓ Novel diverse model selection algorithms at Edge Gateways exploiting model statistics delivered by Edge Devices.

✓ **Domain:** Regression Analytics at the Edge with model selection at the Edge Gateway
EG: Edge Gateway
ED: Edge Device
\( f \): Regression Model
\( C \): Data Statistics
Models Diversity
Edge Device Tasks

- Check **familiarity** of a new measurement
- Sliding Window of recent measurements
  
  \[
  (x, y)_t \quad (x, y)_{t-1} \quad (x, y)_{t-2} \quad \ldots \quad (x, y)_{t-N}
  \]
- Generate Statistics for **Input-Error Space** quantization
- Performing On-line Regression on this window to generate a model $\rightarrow f_i(x)$
- Keeping locally at the Edge Device a copy of the recent model sent to Edge Gateway $\rightarrow f^O_i(x)$
- Update Edge Gateway **only** with updated model parameters w.r.t. statistics of **input-error space**.
Edge Device Familiarity & Input-Error Space Quantization
Edge Device Model Update Mechanism

After identifying Familiarity:

1. Append new measurements in Sliding Window
2. Retrain/adapt model $f_i(x)$
3. Calculate model prediction error with the new model at the Edge Device
4. Calculate model prediction error with the most recent model sent to the Edge Gateway
5. Compare the difference of the prediction errors
   - If absolute difference is above a threshold → Send the new model parameters to Edge Gateway
Edge Gateway Tasks

- Collecting all diverse Models (Model Caching)
- Collecting all Statistics (from Input-Error Space Quantization)
- Receiving regression queries from Cloud/analysts and producing output
- Select the most appropriate **subset** of the cached models
Edge Gateway Model Selection Algorithms

1. Simple Model Aggregation (SMA)
   - Averaging over all predictions $\hat{y} = \frac{1}{n} \sum_{i=1}^{m} f_i^0(x)$

2. Input-space Aware top-K Model (IAM)
   - Selects the model $f(x)$ whose the input prototype is the closest to query q compared to all input prototypes

3. Input/Error-space Aware top-K Model (IEAM)
   - Select the model $f(x)$ whose the input prototype is the closest to query q compared to all input prototypes and best associated performance reflected by the error prototype
Methodologies

1. **Baseline**: Sending continuously data from Edge to Cloud and generate one global model


3. **DPB**: Predict data using linear forecasting models [1]

4. **Model Selection Mechanism** (SMA, IAM, IEAM)

---


Experimental Evaluation

Real contextual data:

- Intel Berkley Research Lab Dataset → 2 Edge Gateways, 25 Edge Devices each (sensors) measuring 3-dim. environmental data (humidity, temperature, light)

Queries:

- Last 120 measurements of each Edge Device → 3000 total

Evaluation Metrics:

1. **Communication**: number of messages sent from EDs to EG
   - Percentage of remaining communication w.r.t the baseline solution

2. **Analytics quality at the Edge Gateway**
   - Regression performance discrepancy w.r.t. ground truth
   - Root Mean Squared Error (RMSE)
   - Mean Absolute Error (MAE)
Efficiency: Communication vs. Analytics Error

Results
Efficiency: Communication vs. Analytics Error

\[ \gamma: \text{fraction of the error difference median} \]
THANK YOU!

Natascha Harth: n.harth.1@research.gla.ac.uk
Christos Anagnostopoulos: Christos.Anagnostopoulos@glasgow.ac.uk
Essence: Pervasive & Distributed Intelligence: http://www.dcs.gla.ac.uk/essence/