Virtual Things for Machine Learning Applications

Gérôme Bovet\textsuperscript{1, 2} – Antonio Ridi\textsuperscript{2, 3} – Jean Hennebert\textsuperscript{2}
\textsuperscript{1}Telecom ParisTech France
\textsuperscript{2}University of Applied Sciences Western Switzerland
\textsuperscript{3}University of Fribourg Switzerland

gerome.bovet@hefr.ch
Data-driven machine-learning techniques are increasingly used for analyzing sensor data
- Trained models performing high accuracy
- Executed server-side

Sensing devices are empowered with high computational capabilities
- Often underexploited CPU and memory

Sensor networks are following the Web-of-Things paradigm
- Strong interaction style between heterogeneous devices
Motivations

- Congestion
- Single-point of failure
- Entry point for attacks

3rd party provider

Runtime

Privacy

Internet

Border Router

Virtual Things for Machine Learning Applications
Discriminative models
- Tree-based decision models
- Simple to implement
- Require no particular capabilities
Machine Learning - Approaches

- **Generative approach**: separately model class conditional densities and priors
  \[ p(x|C_k), \quad p(C_k) \]
  then evaluate posterior probabilities using Bayes’ theorem
  \[ p(C_k|x) = \frac{p(x|C_k)p(C_k)}{\sum_j p(x|C_j)p(C_j)} \]

- **Discriminative approach**: directly model posterior probabilities
  \[ p(C_k|x) \]
Machine Learning - Comparison

**Generative model**
- New classes can be added without re-training using all the data
- Training converges faster
- Need to compute likelihood for each class

**Discriminative model**
- Very fast once trained
- Training converges slower
- Need to re-train with all the data when adding/removing classes

Probability density functions as function of $x$ vs. posteriors as function of $x$. $p(x|C_1)$ and $p(x|C_2)$ for the generative model, and $p(C_1|x)$ and $p(C_2|x)$ for the discriminative model.
Virtual Things for Machine Learning Applications

Architectural Design - Objectives

- Perform machine-learning within the sensor network
- Reuse computational capacities of already installed devices (sensors and actuators)
- Inherit and extend key properties of the Web
  - Strong interaction style
  - Independence regarding hard-/software platforms
  - Scalable architecture
- Formalize the exchange of machine-learning models
- Extend the perception of things to virtual system components
Architecture

Enterprise Network

Learning System

Virtual Sensor

Configuration
(Model deployment)

Runtime
(Class management)

Sensor Network

Virtual Class

Configuration

Class

Virtual Class

Configuration

Class

Likelihood

Model

Measures

Client

Virtual Things for Machine Learning Applications

CoAP

HTTP
Virtual Class

- Preloaded agent with runtime algorithm (HMM or GMM)
  - Discovery by location and available capabilities
- Represents a single class
  - Output is the likelihood for the class (probability)

Virtual Class

Deploying models in JSON and binary

Observable resource returning the current probability
Virtual Sensor

- Represents a high-level sensor (non-physical)
  - Extracts knowledge from multiple sensors of different nature
  - Machine learning tasks abstraction level
  - Reusable component for performing mashups (semantic description of the sensor)

Entry point for deploying machine learning models
Performs validation of the MaLeX input (JSON schema)
Distributes the classes on nearly located agents
Performs the decision making (class with highest likelihood)
MaLeX - Machine Learning Exchange Format

- Inspired from PMML - Predictive Model Markup Languages (no generative models)

- Formalized as JSON schema

- Description of general entities
  - Location, dimensions, type of sensor

- Description specific to HMM and GMM
  - Algorithm type, normalisation, states, matrices (mu, sigma, weights, transition)

- Extensible to other kind of models
Deployment

MaLeX

General properties
- Model 1
- ...
- Model N

Semantic description

Virtual Sensor

Virtual Class

Virtual Class

Numerical Analysis Software

JSON

Mcast GET: D dimensions, N states, K gaussians

2.05 Content: Location

Determine distance from dimensions

PUT: deploy model

General properties

JSON

Model parameters

Binary
Applications - Building Automation Mashup Editor
Applications - Building Automation Mashup Editor
Applications - Building Automation Mashup Editor

Scenario name: Reading
Scenario description: Light when reading in the living room

When this stuff happens... ...then do these things.

- Switch Light
  Kitchen
  - OFF

- Switch Light
  Living room
  - 50%

- Switch Blinds
  Living room
  - Closed

[buttons: Save scenario, Export as RDF file]
From devices to building automation
Performance - Appliance recognition

Virtual Sensor: Raspberry Pi

Virtual Classes: 5x OpenPicus Flyport Wi-Fi PRO

![Graph showing scalability test with concurrent clients.]

![Graph showing round-trip time depending on the operation strategy.]

Success Rate of Requests [%]

Concurrent Clients (5 Requests/Client)

Round-Trip Time Depending on the Operation Strategy

Sync-based and End-to-end strategies are compared.

Virtual Sensor: Raspberry Pi

Virtual Classes: 5x OpenPicus Flyport Wi-Fi PRO

08.10.14

Institut Mines-Télécom

Virtual Things for Machine Learning Applications
Contributions

- **Agent-based HMM and GMM virtual classes in C**
  - Deploy complex models (multidimensional matrices) on constrained devices

- **MaLeX - Machine Learning Exchange Format**
  - Schema describing generative algorithms
  - Embeddable into matlab to automatically export the models in the right format

- **BAME - Building Automation Mashup Editor**
  - Easily integrate new devices by relying on semantic descriptions
Conclusion and future works

**Conclusion**
- Sensing devices are powerful enough for running HMM
- Reusable architecture based on RESTful APIs and semantics
- Building automation machine learning mashups feasible
- No dedicated infrastructure is required for data-driven analysis

**Future works**
- Finding efficient training algorithms segmenting the training data
- Representing training algorithms as Web agents with semantic descriptions
- Distributing training agents within the sensor network
Questions