Demo: Probabilistic Predicates to Accelerate ML Inference

Microsoft Research — Yao Lu, Srikanth Kandula, Surajit Chaudhuri — University of Washington

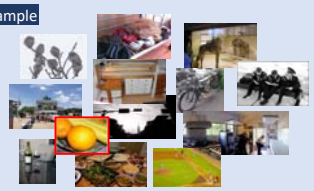
Motivation

Spark SQL, Apache Pig, Hive, Azure Cosmos DB, Myria

Context: ML serving + Relational big-data platforms

SQL + UDFs on images, text, videos…

Example



```
SELECT ImageID FROM UDF_YOLOv2(Images)
WHERE has_orange AND has_banana;
```

Query is slow because each image → UDF.

Speedup? Images → UDF_YOLOv2 → $\sigma_{\text{orange}\wedge\text{banana}}$ → Result

Predicate pushdown? X — No "orange" column here

Pre-materialize? X — Too many UDFs & predicates for ad-hoc queries; high storage cost

Probabilistic Predicates (PPs) [2]:

Input → $PP_{\text{orange}\wedge\text{banana}}$ → YOLOv2Detector → σ

Can offer sizable speed-up if:
- relative cost of PP is small
- achieves large data reduction
- meets target accuracy

Technical problems addressed

- Train PPs for diverse inputs
  - SVMs, KDE, shallow NNs…
  - model selection.
- Avoid per-query training
  - train PP for *simple preds.*,
  - use QO to build PP comb.
- Sizable gains on images, video and text datasets

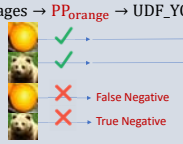
Interactive Demo

Probabilistic Predicates

PPs are **fast** *predicate-specific* filters on input; they reduce work for expensive UDFs.

Example query: retrieve images containing "oranges":

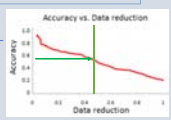
Images → PP_{orange} → UDF_YOLOv2 → σ → Result



True Positive, False Positive, False Negative, True Negative

No harm on query accuracy, but less speedup

Accuracy, Data reduction, Execution cost

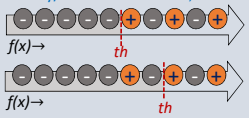


Simple PP using linear SVM:

$PP_{\text{orange}}: f(x) = w \cdot x + b$

PP discards $f(x) \le \text{th}$

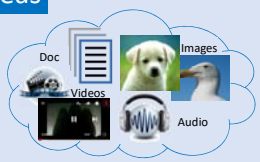
Accuracy/ reduction tradeoff; choose threshold (*th*)



Accuracy = 3/3, Reduction = 5/10

Accuracy = 2/3, Reduction = 7/10

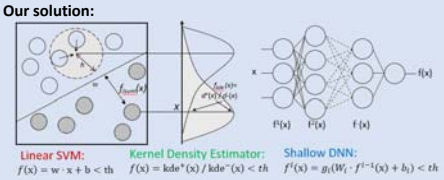
Key ideas



Complex query predicates:
- Large space of possible predicates
- Costly to train/store PP per predicate
- Indiv. PPs do not generalize

Challenge 1: Build PPs for arbitrary inputs ? Dimensionality, Sparsity. Linear separability? …

Our solution:



Any classifier that fits $f(x) < \text{th}$ can work for PP

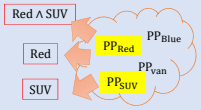
\+ Pre-process: dim. reduction (PCA, feature hashing.) (Opt.)

\+ **Model Selection**

Challenge 2: Use PPs for complex predicates w/o per-query training (i.e., small overhead)?

5 cols x 10 values ⇒ >10^5 predicates — $PP_{\text{red}\wedge\text{SUV}}$ will not work for σ_{SUV}

(i) Build PPs for simple predicates; QO identifies best necessary condition



PPs trained << # predicates

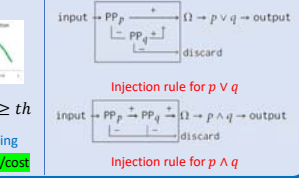
(ii) Assign accuracy budget per PP

$\max_{th_{\text{cat}}, th_{\text{dog}}} \frac{r_{\text{cat}\wedge\text{dog}}}{c_{\text{cat}\wedge\text{dog}}}$, s.t. $a_{\text{cat}\wedge\text{dog}} \ge th$

Solved using dynamic programming

Find the plan with best reduction/cost

(iii) Assemble PPs into plan



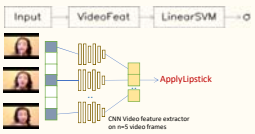
Machine Learning + Query Processing

1. COCO Image Dataset

Input: images
UDF: YOLOv2 Object Detector
Output: [0,1] vector ⇒ Multi-Class labels

```
SELECT img_id FROM YOLOv2()
WHERE has_dog & has person &
      has_car | has_bicycle;
```

2. UCF101 Video Dataset

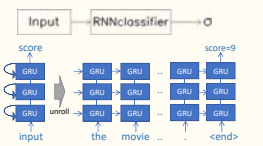


Input: Videos
UDF: CNN Video Features, Linear SVM
Output: category – Single-class label

```
SELECT video_id FROM LSVM()
WHERE cat.= "ApplyLipstick";
```

3. IMDB Movie Review Dataset

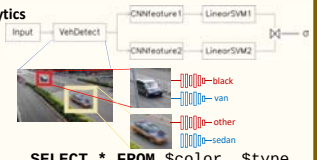
"I came in in the middle of this film so I had no idea about any credits or even its title till I looked it up here, where I see that it has received a mixed reception by your commentators…"
"From the beginning of the movie, it gives the feeling the director is trying to portray something, what I mean to say that instead of the story dictating the style in which the movie should be made…"



Input: movie review text
UDF: Gated-Recurrent-Unit (GRU) RNN
Output: score – [1,10] real value

```
SELECT docId FROM RNNClas.()
WHERE score < 2 | score > 8;
```

4. DETRAC Traffic Surveillance Video Analytics



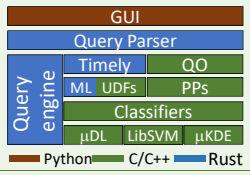
Input: Video frames
UDF: YOLOv2 vehicle detector
For each detected vehicle patch:
CNN Image feature extractor #1,2
LinearSVM classifier #1 ⇒ Veh Color
#2 ⇒ Vehicle Type
Output: veh_color, veh_type

```
SELECT * FROM $color, $type
ON $color.{fid,vid}=
$type.{fid, vid}
WHERE veh_color = red &
veh_type = SUV;
```

System & Demonstration

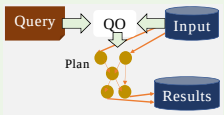
A cross-platform, data-parallel engine for DNN-enabled ML

User query: Standard SQL + UDF syntax. See [Lu. et al. SoCC16].
Timely Dataflow: Rust impl. of Naiad [Murray et al. SOSP13]
Query engine: Join node, filter logic, columnar store for blobs etc.
µDL: A light-weight DNN engine in C/C++.
µKDE: A Kernel Density Estimator in C/C++.



Demonstration Overview
(1) ML query processing with/without PPs. **(2)** User-specified filtering accuracy. **(3)** QO with available PPs

Demonstration Systems:



a) Baseline System w/o PPs

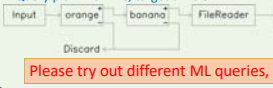
b) Constructing PPs. (Not demonstrated)

c) Full system w/ PPs

Example:
Images where has_orange & has_banana
Original ML query plan:

Candidates plans:
[has_orange&has_banana]: 0.974 *
[has_orange] 0.939, [has_banana]: 0.873

Query plan with PPs, target a = 0.9:



Please try out different ML queries, with different predicates, PPs on/off, and different target accuracies.

References
[1] https://www.reportlinker.com/p04519487/Machine-Learning-as-a-Service-MLaaS-Global-Market-Outlook.html
[2] Y. Lu, A. Chowdhery, S. Kandula, S. Chaudhuri. Accelerating Machine Learning Inference with Probabilistic Predicates. SIGMOD 2018.
[3] Y. Lu, A. Chowdhery, S. Kandula. A Relational Platform for Efficient Large-Scale Video Analytics. SoCC. 2016.
[4] D. Murray et al. Naiad: a timely dataflow system. *SOSP 2013.*