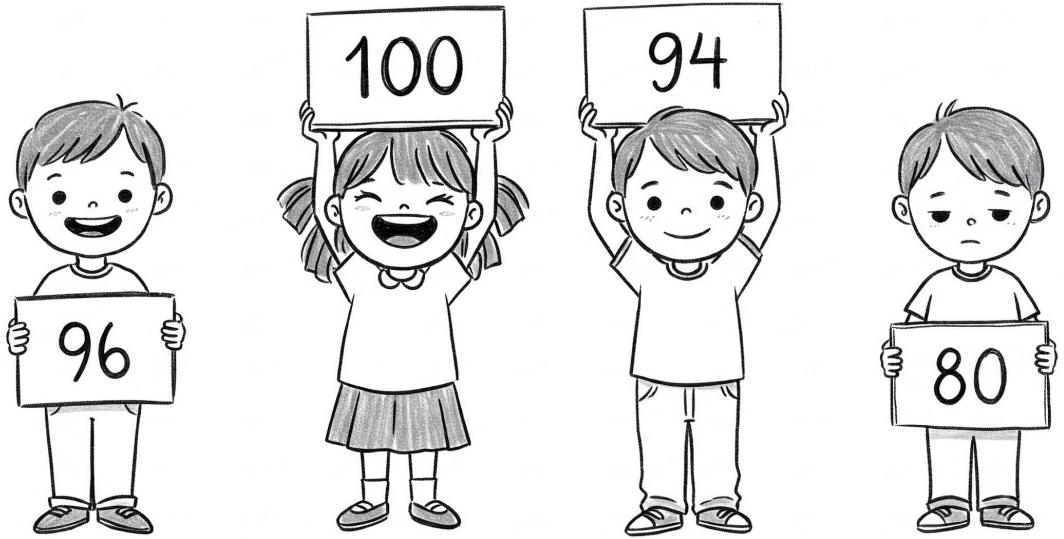
Real-Time Anomaly Detection with Apache Beam

Shunping Huang

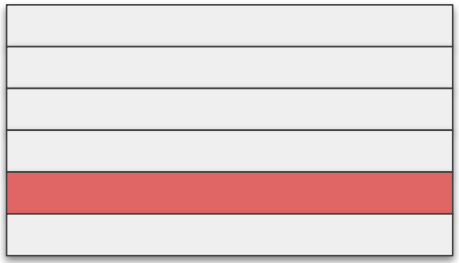
Anomaly Detection

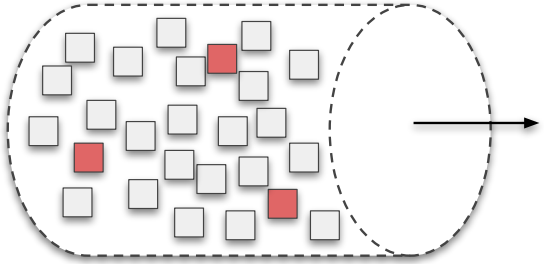


- **Anomaly**: the data instance that is **different** from the **normal** ones.
- **Anomaly Detection**: the task to identify the anomalies.

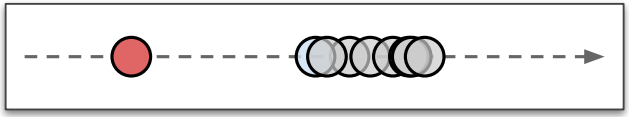
Anomalies in Different Types of Data

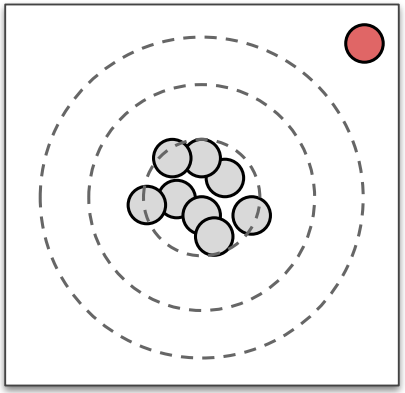
- **Bounded** vs. **Unbounded**





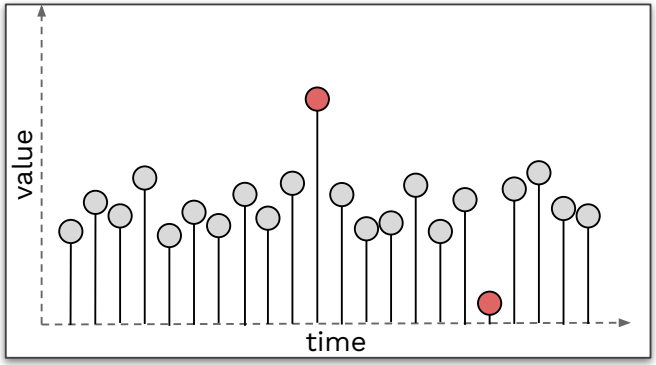
- **Univariate** vs. **Multivariate**

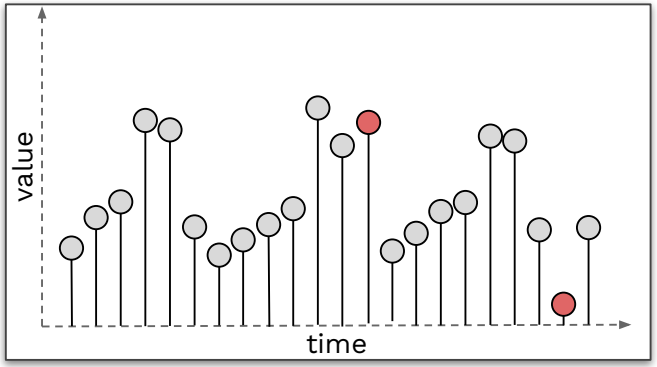




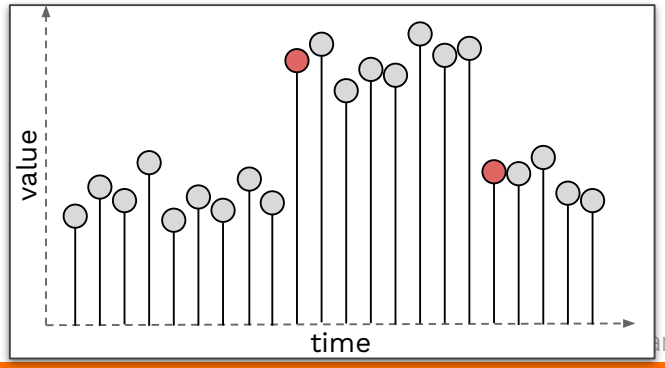
Anomalies in Different Types of Data (cont.)

- **Data Streams** vs. **Time Series**



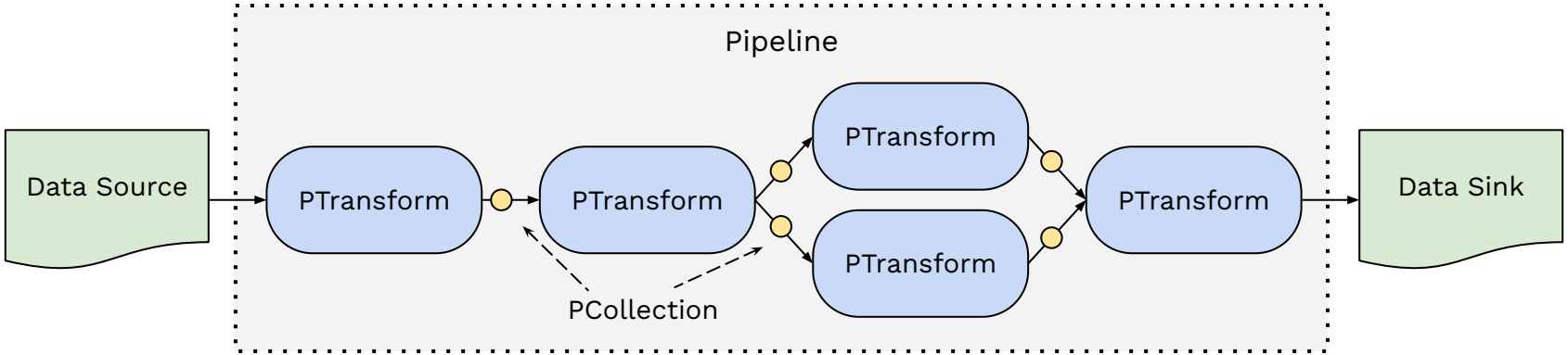


- Data Streams with **Concept Drift**

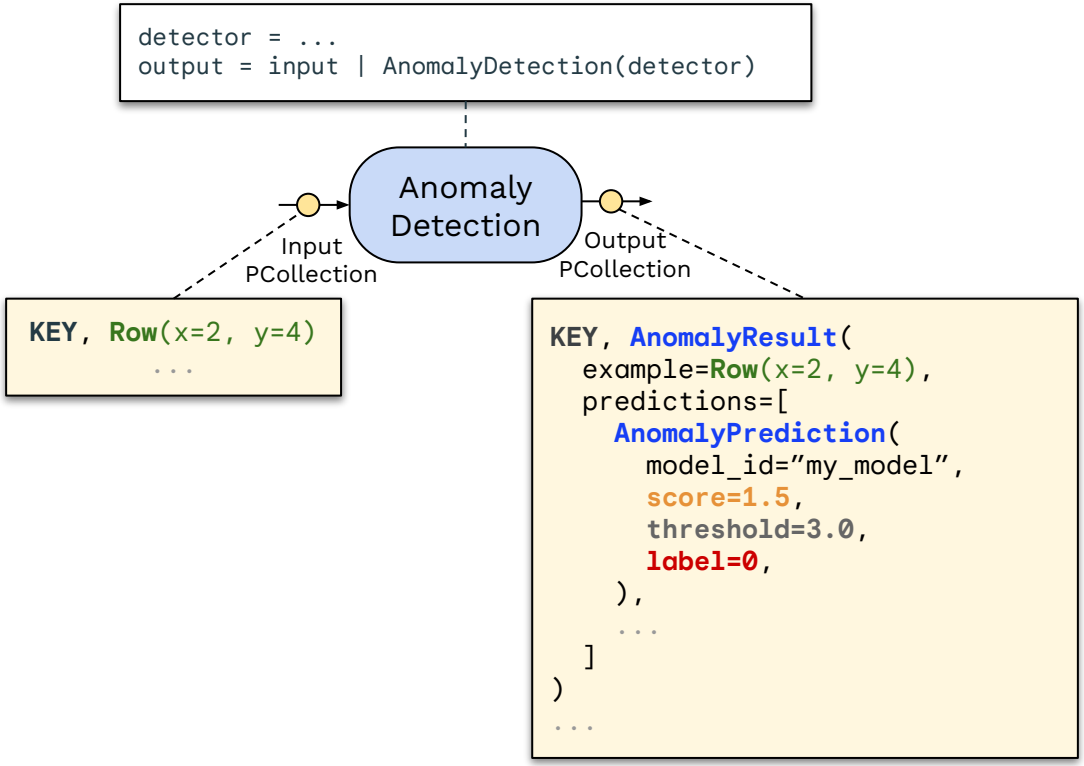


One-Pager: Beam Basics

- A framework to unify the **batch** (bounded) and **streaming** (unbounded) processing.
- Key Concepts
  - **PCollection** - a representation of data for parallel processing
  - **PTransform** - a representation of computation to transform data
  - **Pipeline** - a Directed Acyclic Graph (DAG) of PTransforms

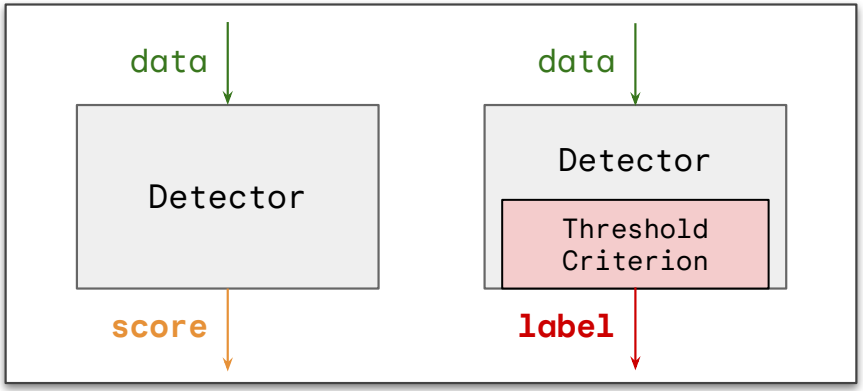


PTransform for Anomaly Detection



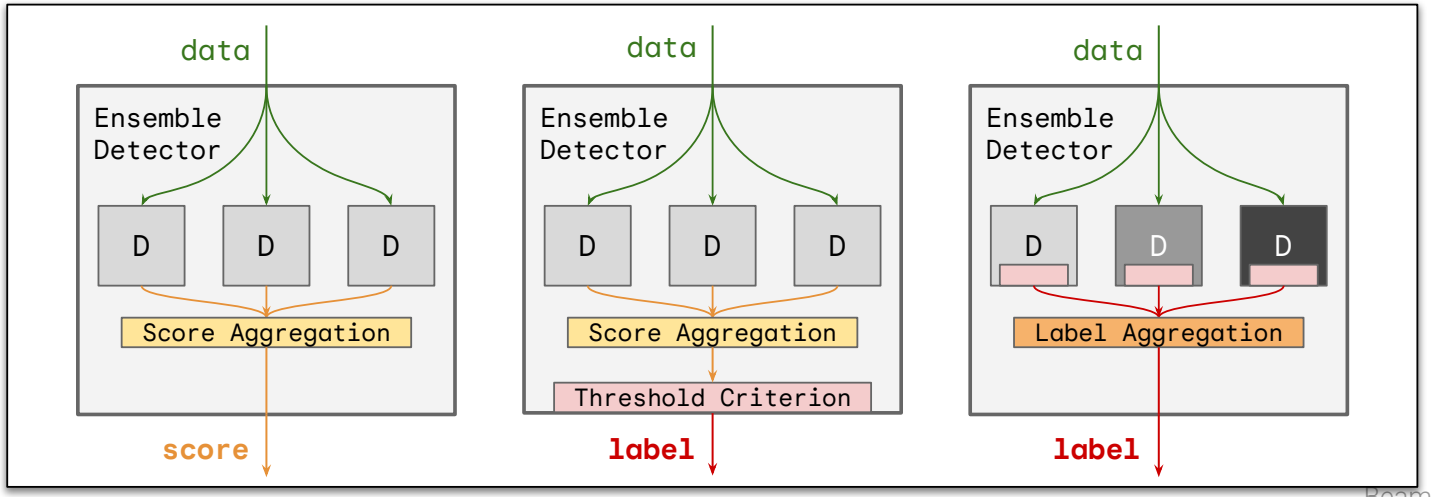
Internals

- **Individual Detectors**
  - Run a specific anomaly detection method and generate scores or labels
  - e.g. Incremental Z-Score, IQR, and offline models (such as isolation forests, LOF, one-class SVM, etc) supported by PyOD.
- **Threshold Criteria**
  - Fixed Thresholding, Quantile Thresholding



Internals (cont.)

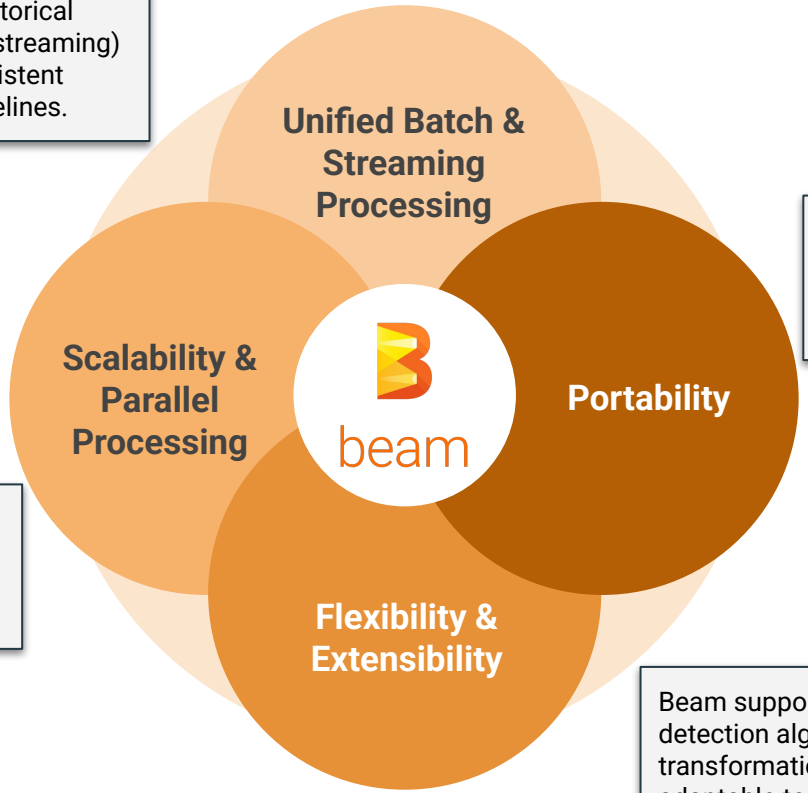
- **Ensemble Detectors**
  - Run a set of sub-detectors in parallel and aggregate scores or labels
- **Aggregation Strategies**
  - For Scores: Average Score, Max Score
  - For Labels: Majority Vote, All Vote, Any Vote



Demo

Key Advantages of Beam for Anomaly Detection

Beam handles both historical (batch) and real-time (streaming) data, allowing for consistent anomaly detection pipelines.



Beam pipelines can run on multiple execution engines (runners) like Apache Flink and Google Cloud Dataflow.

Beam's distributed processing capabilities enable efficient handling of large datasets and high-velocity data streams.

Beam supports various anomaly detection algorithms and custom transformations, making it adaptable to diverse needs.

Links

- **Design Doc**
  https://docs.google.com/document/d/1tE8lz9U_vjlNn2H7t-GRrs3vfhQ5UuCgWiHXCRHRPns/edit?usp=sharing
- **Source Code**
  https://github.com/apache/beam/tree/master/sdks/python/apache_beam/ml/anomaly
- **Python Doc**
  https://beam.apache.org/releases/pydoc/current/apache_beam.ml.anomaly.html
- **Colabs**
  https://github.com/apache/beam/tree/master/examples/notebooks/beam-ml/anomaly_detection

Thank you!

Try our new Apache Beam anomaly detection and give us feedback.
Contributions are very welcome!

Shunping Huang
shunping@google.com
https://www.linkedin.com/in/shunping-huang-545b0150