Karlsruhe Institute of Technology

Masterthesis

Change-Adaptive Active Learning on Data Streams

Julien Aziz

Motivation - Active Learning & Data Streams

Active Learning

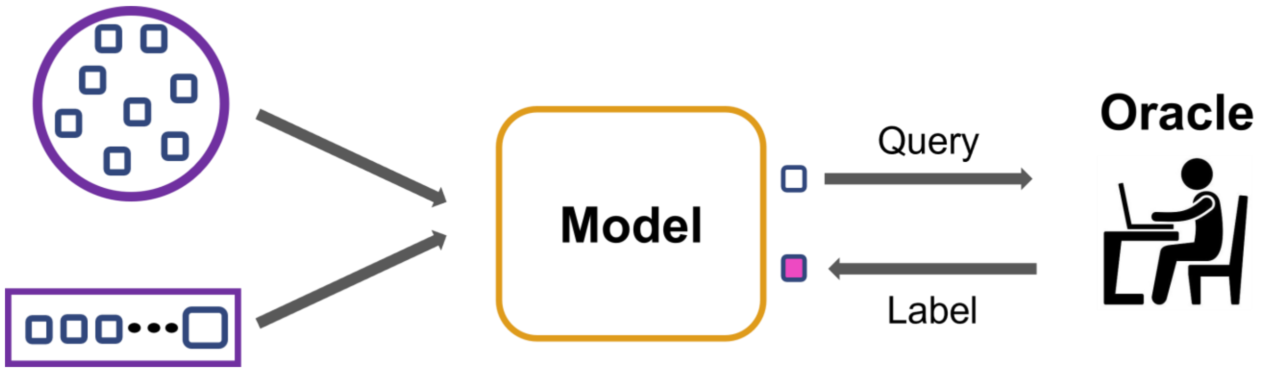
- Model Accuracy vs. Labeling Costs
- Accurate classifier with less training
- Algorithm chooses instances to label
- Main aspects:
  - Input scenario
  - Labeling strategy
  - Labeling budget

Data Stream Mining

- Time-sequenced streams of data
- Rising data volume and arrival rate
  - Labeling every instance unfeasible
  - Active Learning to maximize accuracy
- Main aspects
  - Time requirements
  - Memory requirements
  - Change adaption

Motivation – Active Learning

- Main Challenge: find most valuable training instances (Labeling Strategy)
- Labeling Strategy assesses instances
  - **Uncertainty Based**: Labels around decision boundary
  - **Representation Based**: Labels reflect feature space distribution
- Only “valuable” instances presented to Oracle
  - **Pool-based**
    - Entire dataset available
    - Instance ranking
  - **Stream-based**
    - Instance at a time
    - Threshold comparison



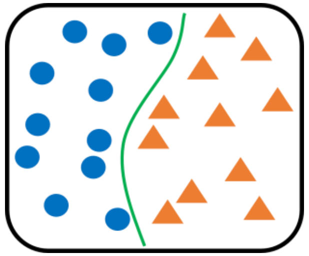
Motivation – Data Stream Active Learning

- Data Stream $D = \ \dots, (X_{t-1}, y_{t-1}), (X_t, y_t), \dots$
  - $X_t \sim p_t(X)$: feature vector
  - $y_t \sim p_t(y)$: class label
  - (X_t, y_t) is sample from joint distribution $p_t(X, y)$
- Label selection
  - Batch-processing: Pool-based techniques on stream batches
  - Instance-incremental: Decision on each instance at a time
- Challenges:
  - Thresholds must be constantly adjusted
  - Concept drifts

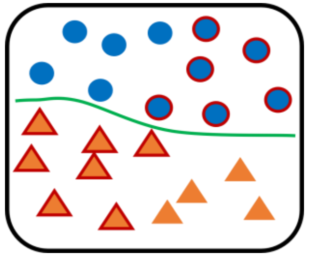
Motivation – Concept Drifts

Karlsruhe Institute of Technology

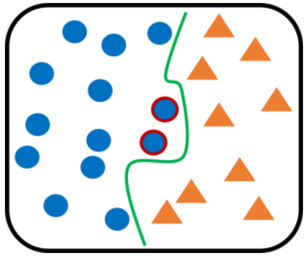
- Data changes over time: $p_t(X, y) \neq p_{t+1}(X, y)$
  - Caused by change in $p_t(X)$, $p_t(y)$ or $p_t(X,\ y)$
  - Acquired Labels may become outdated
  - Different types possible:
    - Original



    - Global



    - Regional



- Further differentiations:
  - Abrupt
  - Gradual
  - Reoccuring

- Reacting to changes challenging with Active Learning
  - Data is sparsely labeled
  - Labeled instances not representative

Related Work – Addressing Changes

Karlsruhe Institute of Technology

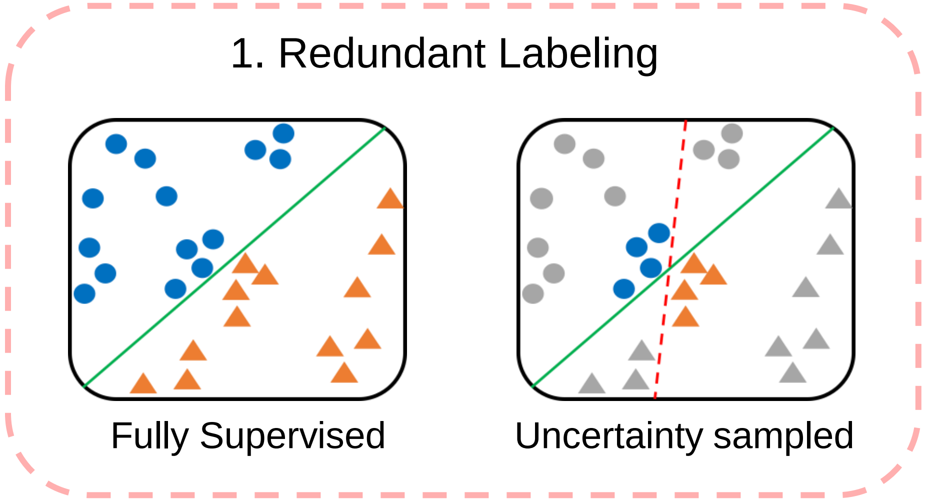
- Implicit Change Adaption
  - Dual Query & Cognition Window [Lui et al. – 2021]
  - *ACLStream*: Clustering-Based [Lenco et al. – 2018]
- Periodically update the model
  - Batch-Incremental Methods [Zhu et al.- 2007]
  - Window-Based Methods [Kottke et al. – 2015]
- Explicit Change Detection Frameworks
  - AL Framework for Data Streams [Žliobaitė et al. – 2014]
  - PEFAL: Extention of Žliobaitė et al. (2014) [Xu et al. – 2016]

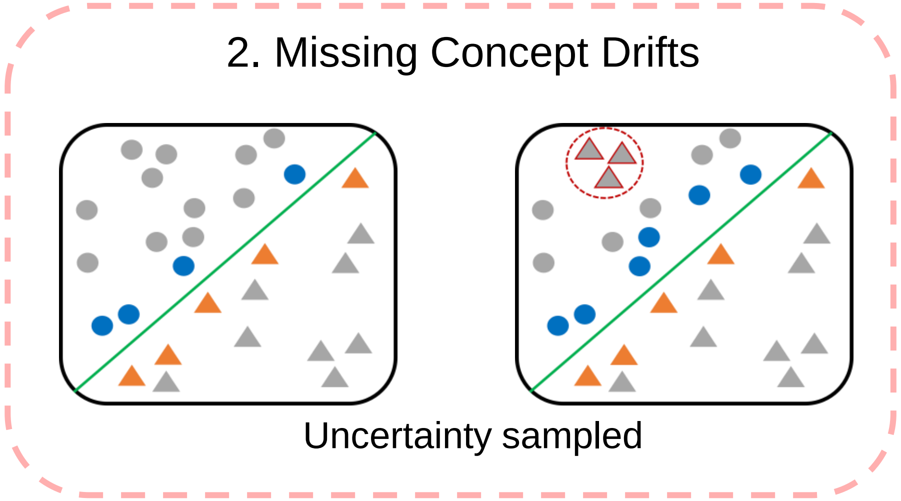
- **Problems**:
  1. Overadaption of samples around decision boundary [Zhang et al. – 2019]
  2. Tradeoff: Discard potentially valuable information / Consider old concepts [Gama et al. 2014]
  3. Delayed reaction to abrupt changes [Lui et al. – 2021]

Illustration – Overadaption

1. Overadaption of samples around decision boundary [Zhang et al. – 2019]

- Uncertainty-based strategies acquire labels at decision boundary
  - Most valuable instances to learn the classification problem
  - Problems:





- Balanced Labeling Strategy: Uncertainty + Representation based

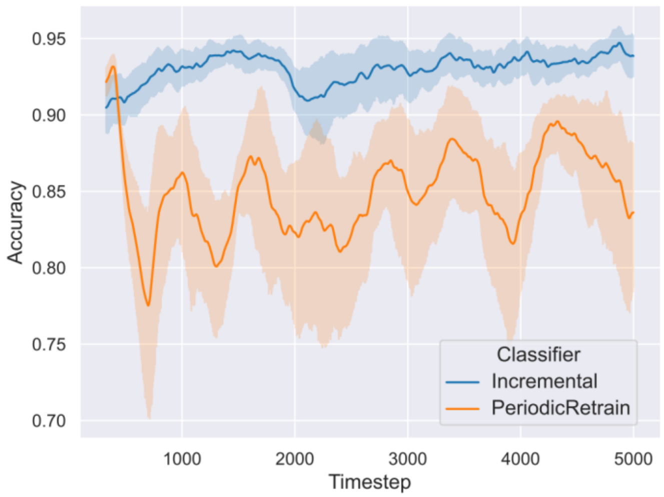
Illustration – Preserving vs. Forgetting

2. Tradeoff: Discard potentially valuable information / Consider old concepts [Gama et al. - 2014]

- Forgetting mechanisms or preserving knowledge
  - Periodic retraining vs incremental training

Experiment Setup:
- Hyperplane Generator
- Length: 5000 Samples
- Classifier: Hoeffding's Trees
- Label Budget: 10%
- Repetitions: 10

Few Changes (1%)

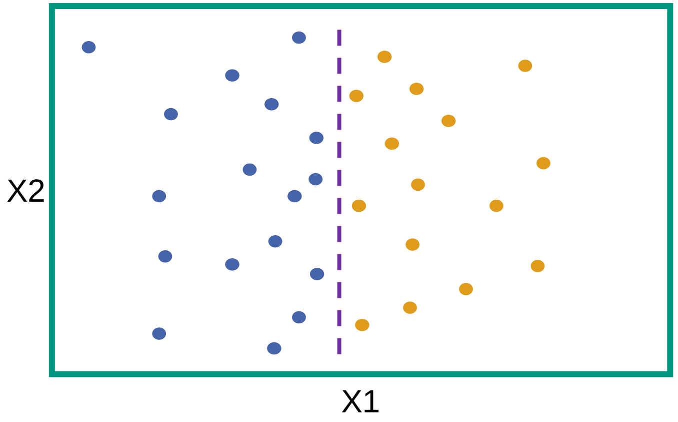


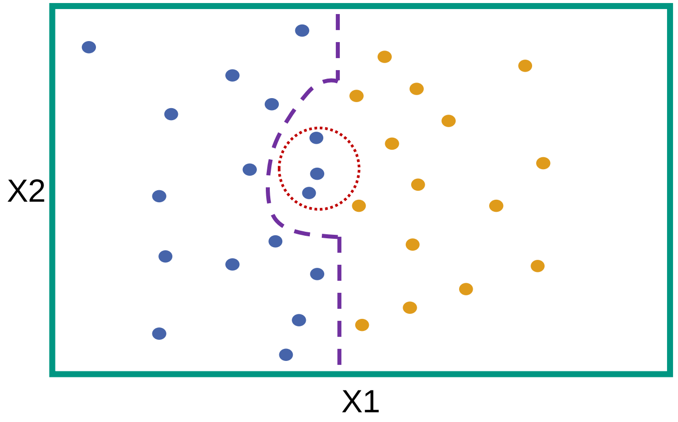
Frequent Changes (30%)

- Explicit Change Detection: Retrain only when changes are identified

Illustration – Regional Changes

3. Delayed reaction to abrupt changes [Lui et al. – 2021]





- Changes can occur everywhere in feature space
  - Decision Boundary might change
- Explicit Change Detection based Approaches
  - Completely discard old information
  - Need to retrain from scratch after change

- Changes might only affect a specific region [Liu et al. 2017]
  - Labels in this Region are outdated
  - Labels outside the Region still yield current concept
- Forget only outdated information
  - Faster recovery from abrupt drifts

Requirements – Adaptive Online AL

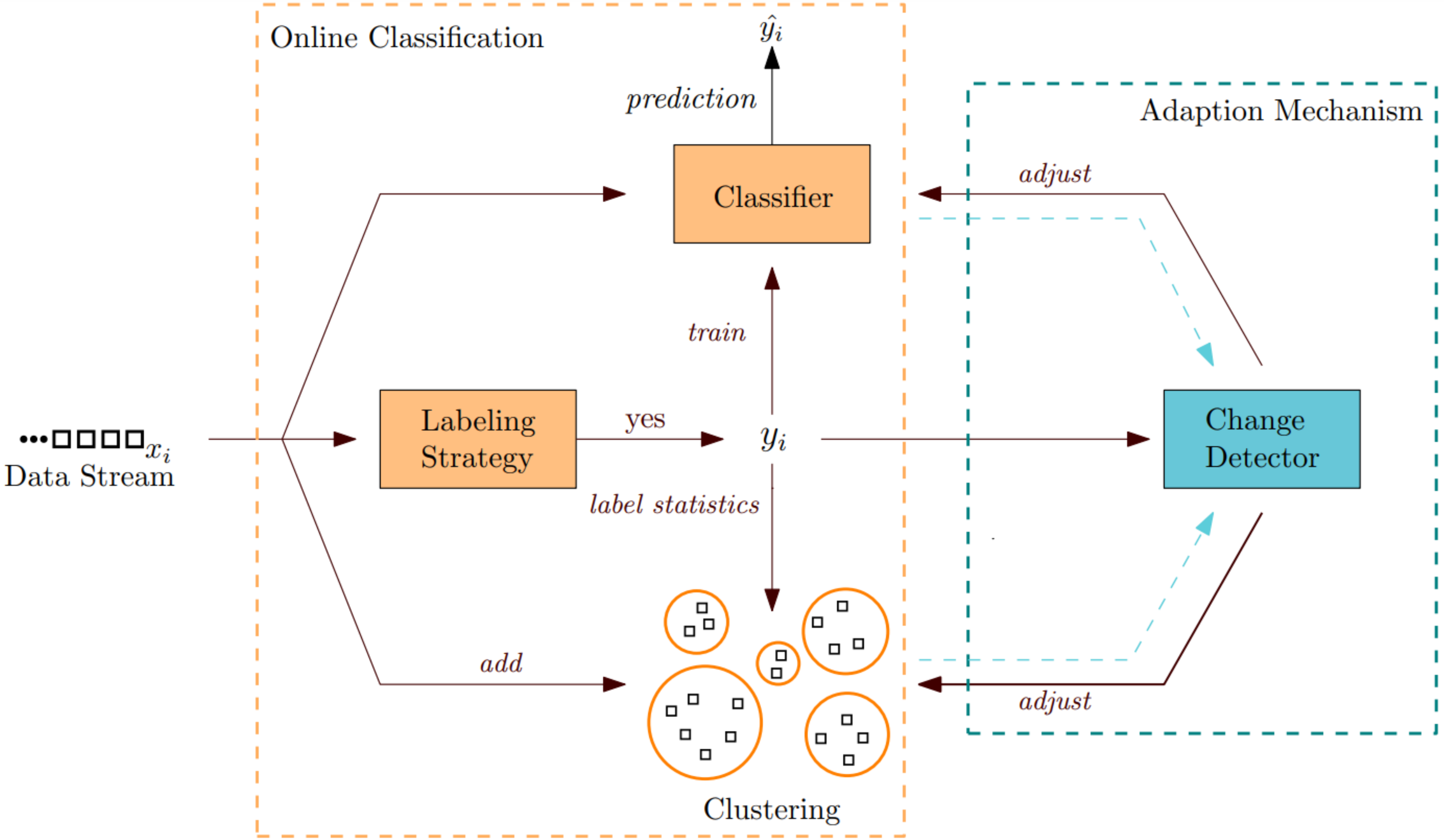
Karlsruhe Institute of Technology

- Requirements for adaptive active learning algorithms in data streams
  - **R1**: Employ balanced labeling strategies (Overadaption)
  - **R2**: Continuously learn in stable stream states (Stability)
  - **R3**: Actively detect concept drifts (Adaptivity)
  - **R4**: Adjust the model by forgetting outdated knowledge (Adaptivity)
  - **R5**: Preserve knowledge unaffected by concept drifts (Recovery time)
- Challenges:
  - Identify the region of changes
  - Adapt the model without losing intact knowledge

Our Approach – CORA

- **C**lustering-Based **O**nline **R**e**A**ctive Learning Framework (CORA)
- Local adaption framework for active learning in data streams
- Main components:
  - **Clustering Algorithm**: Online clustering enriched with labeling statistics
  - **Change Detector Ensemble**: Each monitoring individual cluster
- Interchangable components:
  - **Classifier**: Any incremental trainable probabilistic classifier
  - **Labeling Strategy**: Any online instance-incremental labeling strategy
- 4 Different configurations
  - Single Classifier Model or Ensemble
  - Change Detection on Local Prediction Error or Class Entropy

CORA – Architecture



CORA – Clustering

- Based on CluStream: Unsupervised clustering algorithm for data streams
  - Cluster-Feature: $(LS_i^x, SS_i^x, LS_i^t, SS_i^t, n)$
  - Adaption through merging and deleting clusters based on their relevance
- Enriched Cluster-Feature: $\mathrm{CF_i} = \left(LS_i^x, SS_i^x, LS_i^t, SS_i^t, n, \boldsymbol{LI_i}, \boldsymbol{LD_i}\right)$
  - $\boldsymbol{LI_i} = \{(\boldsymbol{x_0}, \boldsymbol{y_0}, \boldsymbol{i_0}), \dots, (\boldsymbol{x_{w-1}}, \boldsymbol{y_{w-1}}, \boldsymbol{i_{w-1}})\}$
    - Sliding window of recent labeled instances and their timestamps
  - $\boldsymbol{LD_i} = [\boldsymbol{D_i^0}, \dots, \boldsymbol{D_i^{c-1}}]$
    - D_i^j = sum of occurrence of j-th class in cluster i
    - Class Distribution
- Cluster deletions can be triggered externally by change detectors

CORA – Classifier Model

- Base Classifier (M): Any incremental probabilistic classifier
  - Outputs probability score $p_M(y|x_i)$
  - Learns global context
- Clustering Model (C)
  - $p_C(y|x_i)$ approximated using the class distribution $\boldsymbol{LD_i}$ in cluster
  - Captures local classification problem

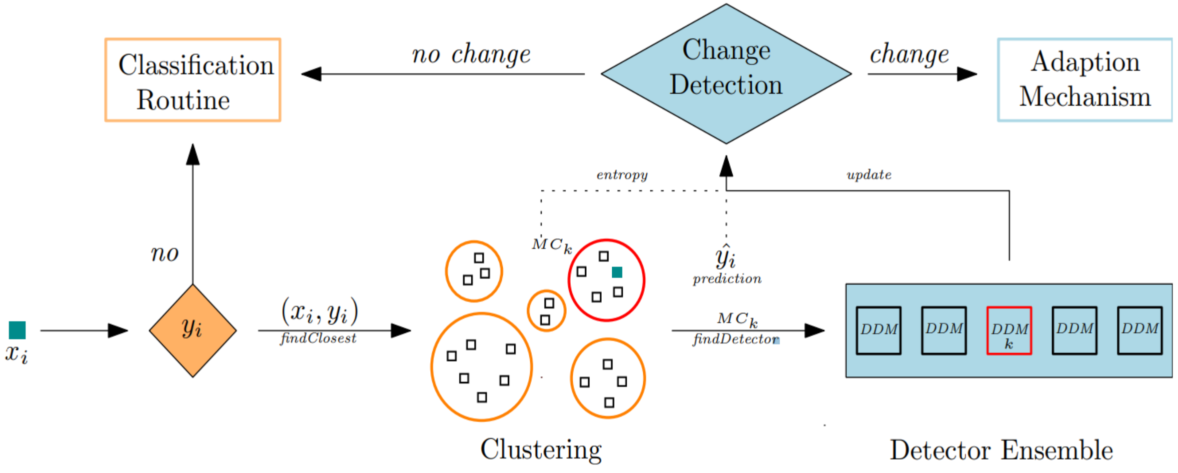
1. Single Classifier Approach

- Base Classifier (M)
- $\hat{y}_i = \arg\max_y p_M(y|x_i)$

2. Ensemble Classifier Approach

- Dual Model:
  - Base Classifier (M)
  - Clustering Model (C)
- $p_E(y|x_i) = 0.5 * (p_M(y|x_i) + p_C(y|x_i))$
- $\hat{y}_i = \arg\max_y p_M(y|x\_i)$

CORA – Change Adaption



Change Detection

- Drift Detection Modules (DDM)
  - Each DDM monitors own cluster
- Procedure
  1. Find closest cluster
  2. Update corresponding detector

Adaption Mechanism

- Procedure
  1. Delete affected cluster
  2. Retrain classifier on remaining cluster
- Recent training data in cluster

- Drift Indicator
  1. **Local Prediction Error**
     - $e_i \begin{cases} 0, & if\ \hat{y}_i = y_i \\ 1, & if\ \hat{y}_i \neq y_i \end{cases}$
  2. **Class Entropy** in cluster
     - $H(CF_k) = -\sum_j^c p_j \log_c p_j$

CORA - Hyperparameters

Detector Threshold δ

- Sensitivity of Change Detectors
- Trade-off
  - Too small – False Alarms
  - Too high – Missed Drifts
- $\boldsymbol{\delta = 1}$ good balance

Number of Cluster $\mathbf{n_c}$

- Influences quality of clustering
  - Predictions and Adaptivity
- Trade-off
  - Too small – Coarse clustering
  - Too high – Sparse cluster
- $n_c = n_{classes} + 8$

Evaluation – Setup

CORA - Configurations

Config	Prediction Model	Detection Indicator
CORA-SP	Single	Prediction Error
CORA-SE	Single	Entropy
CORA-EP	Ensemble	Prediction Error
CORA-EE	Ensemble	Entropy

- Interchangable Components
  - **Hoeffding‘s Trees**
  - **OPAL** [Kottke et al. 2015]
    - Online Probabilist Active Learning

Datasets

Real World					
Dataset	**Features**	**Classes**	**Cat. Feature**	**Length**	**Drift Type**
Electricity	8	2	1	45,312	Unknown
Airlines	7	2	2	539,383	Unknown
Covertype	54	7	44	581,012	Unknown
Pokerhand	10	10	5	829,201	Unknown
Artificial					
Dataset	**Features**	**Classes**	**Cat. Feature**	**Length**	**Drift Type**
Hyperplane	2	2	0	Inf	Increm. + Global
SEA	3	2	0	100,000	Abrupt + Global
RBF	2	15	0	Inf	Abrupt + Local
Chessboard	2	8	0	200,000	Abrupt + Global

Evaluation – Comparison

Baselines

- **OPAL-NA** [Kottke et al. 2015]
  - No active adaption mechanism
  - Balanced Labeling Strategy
- **Zliobaite** [Zliobatie et al. 2014]
  - Explicit change detection
  - Replacement when change detected
- **PEFAL** [Xu et al. 2016]
  - Implicit change detection
  - Two classifier trained parallel
  - Swaped when change detected

Configurations

Baseline	Labeling Strategy	Change Detection	Classifier
OPAL-NA	OPAL + BIQF	-	Hoeffding's Tree
Zliobaite	OPAL + BIQF	DDM	Hoeffding's Tree
PEFAL	VarUncertainty	Implicit	Hoeffding's Tree

- Parameters as in original source

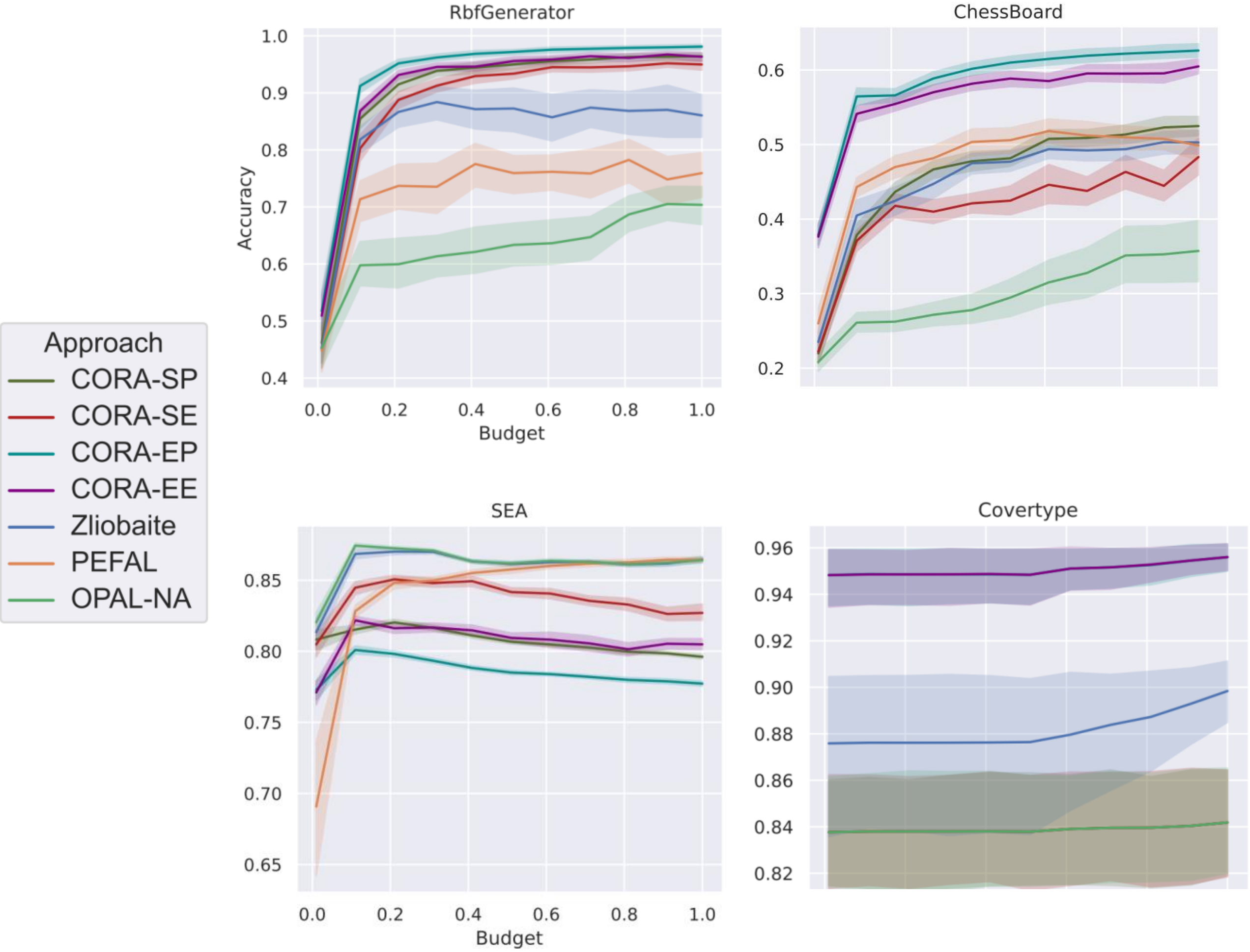
Experimental Setup

- Prequential evaluation
- Stream Partitions: 10 000 Instances
- Budgets: $b \in [0.01, 1]$
- Repetitions: 30

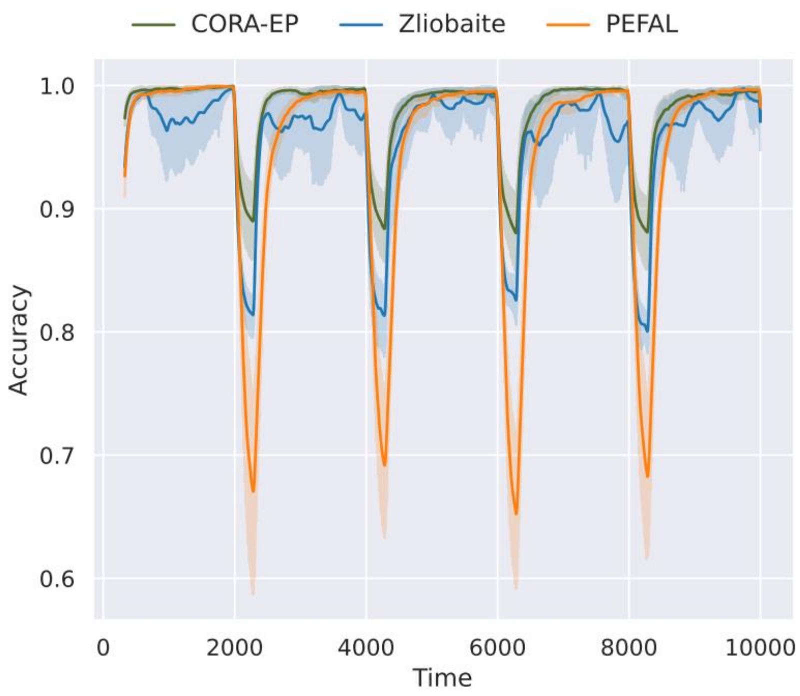
Evaluation – Performance Comparison

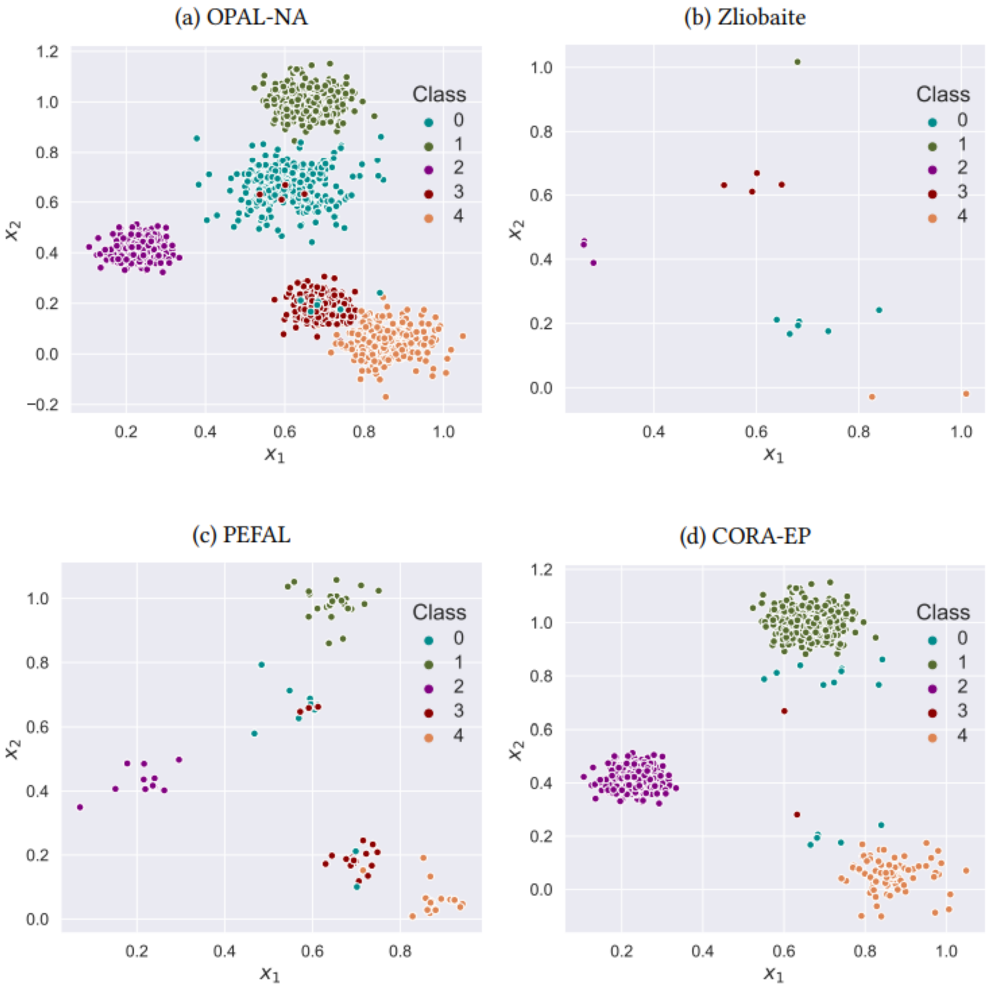
Dataset	CORA-EE	EP	SE	SP	OPAL-NA	PEFAL	Zliobaite
Airlines	0.6157	0.6013	**0.6256**	0.6079	0.6253	0.6086	0.6236
ChessBoard	0.5625	**0.5832**	0.4128	0.4580	0.2981	0.4736	0.4498
Covertype	**0.9506**	**0.9506**	0.8389	0.8389	0.8389	0.8389	0.8817
Electricity	**0.8020**	0.7999	0.7659	0.7491	0.7646	0.7728	0.7932
Hyperplane	0.8536	0.8598	0.8525	0.8619	0.8085	**0.9146**	0.8477
PokerHand	0.7171	**0.724**	0.6361	0.6603	0.6361	0.6715	0.6950
RbfGenerator	0.9064	**0.9251**	0.8778	0.8970	0.6270	0.7253	0.8267
SEA	0.8068	0.7854	0.8364	0.8072	**0.8615**	0.8402	0.8599

- CORA superior on most data sets
- Change detection designs deviate
  - Error based better with sever changes
  - Entropy more robust against noise
- Limitations
  - Categorical attributes with many values
  - High dimensions



Evaluation – Change Adaption





- Faster recovery from abrupt change
- CORA preserves intact knowledge

Evaluation – Requirements

Karlsruhe Institute of Technology

Requirements

- **R1**: Employ balanced labeling strategies (Overadaption)
- **R2**: Continuously learn in stable stream states (Stability)
- **R3**: Actively detect concept drifts (Adaptivity)
- **R4**: Adjust the model by forgetting outdated knowledge (Adaptivity)
- **R5**: Preserve knowledge unaffected by concept drifts (Recovery time)

CORA

- Modular structure
  - Balanced Labeling Strategy
  - Incremental learning Classifier
- Explicit Change Detection
  - Adjust when necessary
- Local Adaption Mechanism
  - Forgetting affected areas
  - Preserving unaffected knowledge

Conclusion & Future Work

- Local Adaption Mechansim
  - Based on clustering
  - Maintain valuable knowledge
  - Faster recovery from local abrupt drifts
- Further usage of Clustering
  - Minimal additional complexity
  - Valuable prediction support
- Future Works:
  - Develop a Labeling Strategy based on our clustering
  - More efficient retraining and merging techniques

References

- Previous Works

  - Zhu, X., Zhang, P., Lin, X., Shi, Y.: Active learning from data streams. In: ICDM. pp. 757–762. IEEE Computer Society (2007)
  - Zliobaite, I., Bifet, A., Pfahringer, B., Holmes, G.: Active learning with drifting streaming data. IEEE Trans. Neural Networks Learn. Syst. 25(1), 27–39 (2014)
  - Liu, Sanmin, et al. "Online active learning for drifting data streams." *IEEE Transactions on Neural Networks and Learning Systems* (2021).
  - Ienco, D., Bifet, A., Zliobaite, I., Pfahringer, B.: Clustering based active learning for evolving data streams. In: Discovery Science. Lecture Notes in Computer Science, vol. 8140, pp. 79–93. Springer (2018)
  - Kottke, D., Krempl, G., Spiliopoulou, M.: Probabilistic Active Learning in Datastreams (2015)

Evaluation – Sensitivity Analysis

Hyperparameters

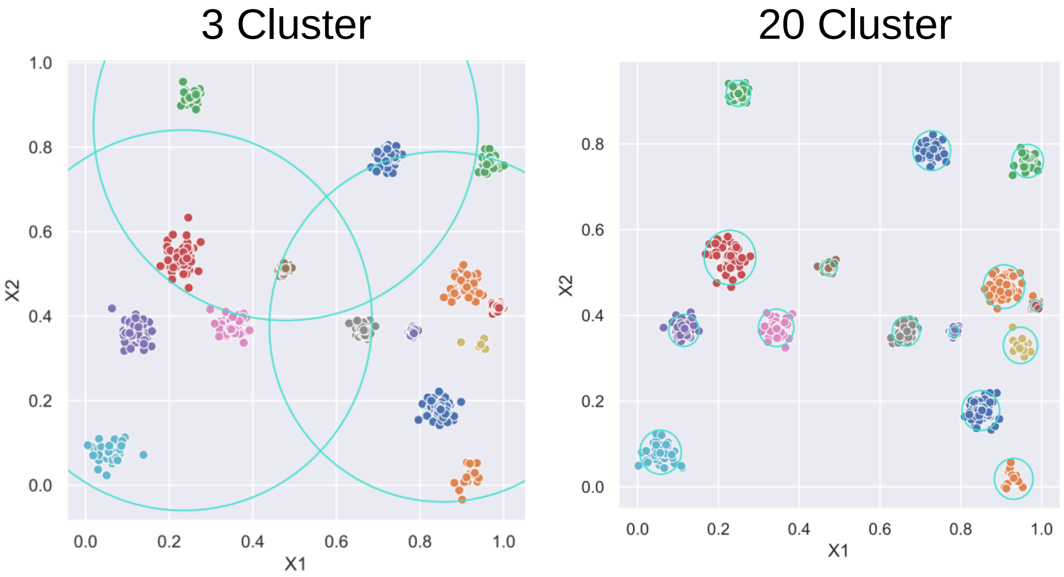
- Detector Threshold $\boldsymbol{\delta} \in [0.1, 3]$
  - Sensitivity of Change Detectors
  - Trade-off
    - Too small – False Alarms
    - Too high – Missed Drifts
- Number of Cluster $\mathbf{n_c} \in [1, 50]$
  - Influences quality of clustering
    - Predictions and Adaptivity
  - Trade-off
    - Too small – Coarse clustering
    - Too high – Sparse cluster

Evaluation

- Hyperplane Generator
  - 2 Classes
  - Hyperplane constantly rotates
  - Incremental Global Drift
- Rbf Generator
  - 15 Classes (15 Rbfs)
  - Random Rbfs swap positions
  - Abrupt Local Drift
- Experiments
  - Partitions: 5000 Instances
  - Budgets: $(10\%, 40\%, 70\%)$
  - Repetitions: 30

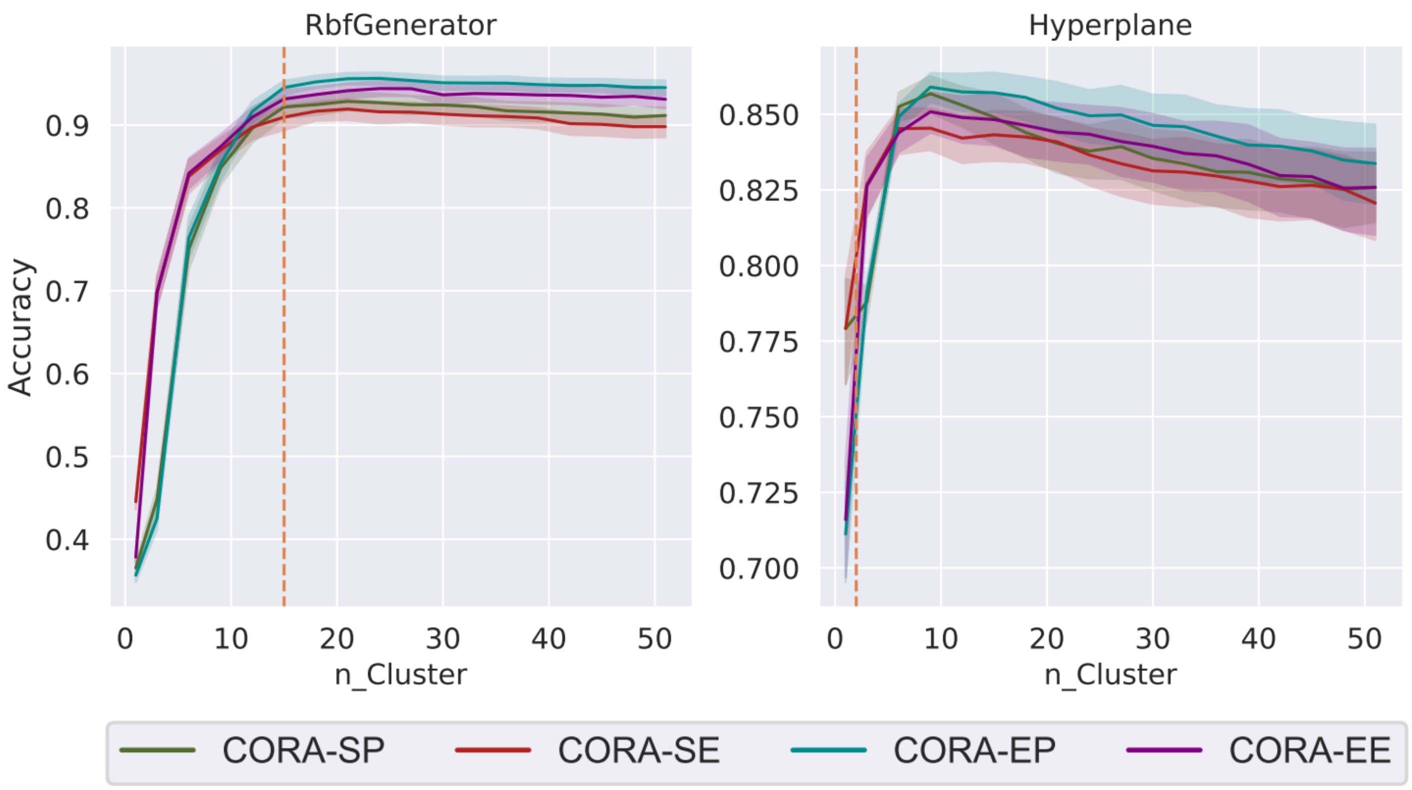
Evaluation – Detector Threshold

- Dependency to number of cluster
  - High sensitivity when small
  - Especially on Rbf Generator
- Entropy based more sensitive



Evaluation – Number of Cluster

- Observation
  - Rise until optimum
  - Degradation beyond
  - Depends on dataset
- Trade-off:
  - Small – Clustering too coarse
  - High – Sparse cluster
- Ensemble more sensitive
  - Direct dependency
- Slightly above number of classes
  - $n_{cluster} = n_{classes} + 8$



Motivation – Stream Active Learning

- Traditional Active Learning assumes stationary relationships between features and target variable
- Data Stream $D = \ldots, (X_{t-1}, y_{t-1}), (X_t, y_t), \ldots$
  - $X_t \sim p_t(X)$: feature vector
  - $y_t \sim p_t(y)$: class label
  - (X_t, y_t) is sample from joint distribution $p_t(X, y)$
- **Concept Drifts**
  - Streams can change over time: $p_t(X, y) \neq p_{t+1}(X, y)$
    - Caused by change in $p_t(X)$, $p_t(y)$ or $p_t(X,\ y)$
    - Acquired Labels may become outdated
    - Model trained on old concept
  - Reacting to changes challenging with Active Learning
    - Data is sparsely labeled
    - Labeled instances not representative

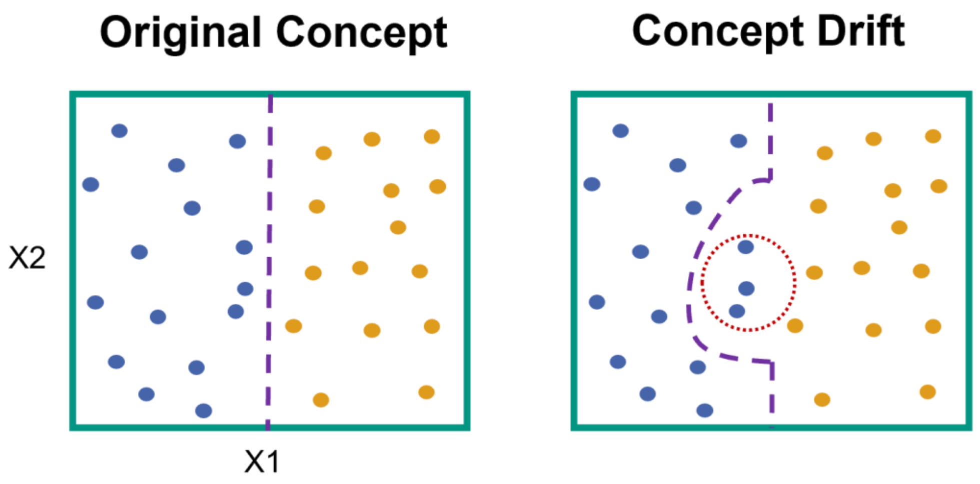
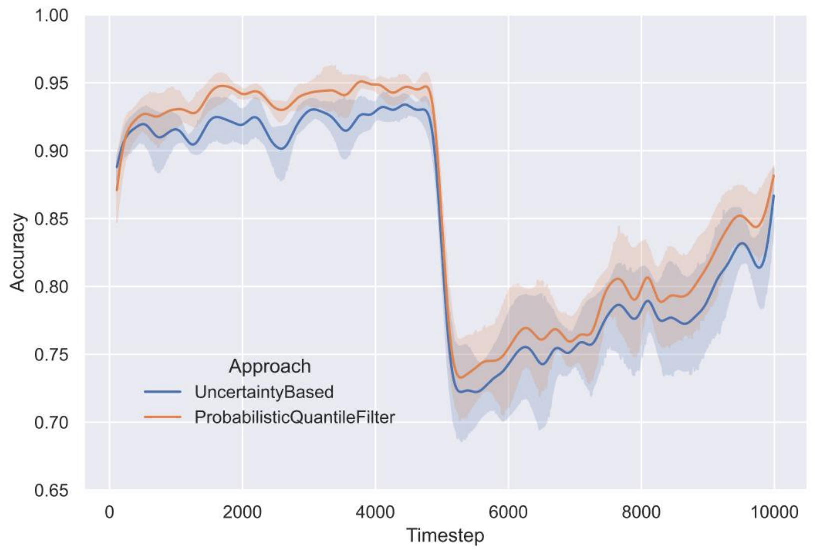
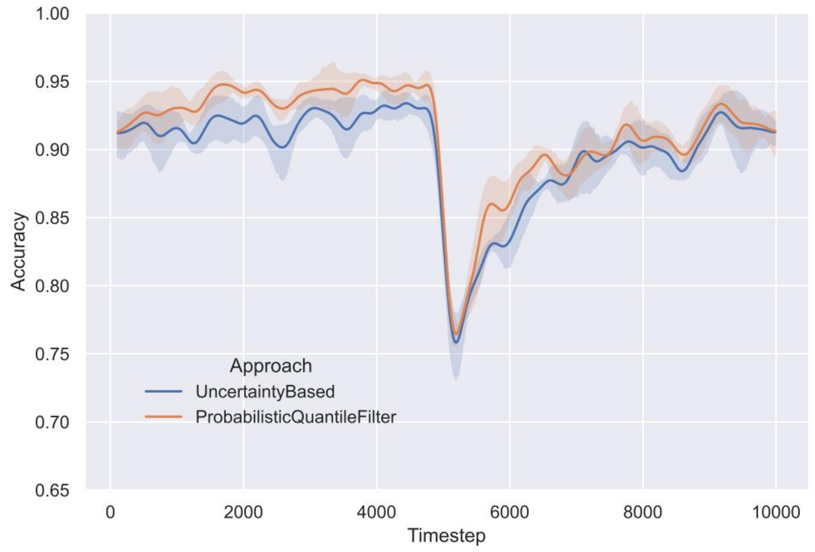


Illustration – Influence of changes on incremental classifier

Incremental Training



Retraining



Experiment Setup:

- Data : Hyperplane Dataset
- Classifier: Incremental
- Label Budget: 10%

- Common approaches fail to adapt to Abrupt Drifts
  - Fix labeling budget prevents fast adaption
  - Predictions after change still based on old concepts
- Detecting changes early allows acting appropriately
  - Storing instances once a Concept Drift occurred
  - Retrain model on new concept

Our Idea

- Detect changes early: Based on labeled & **unlabeled** data
  - In any region of the feature space
- Differentiate between states of the Data Stream
- In **Stationary states**:
  - Preserve labeling budget
  - Discard irrelevant information
- In **Drift states**:
  - Use labeling budget
  - Discard old labels in change region
  - Obtain labels in region of the drift
- Meet Label Budget: Number of labels average the pre-defined budget
- Main Challenge: How to detect regional changes early in Active Learning scenario?

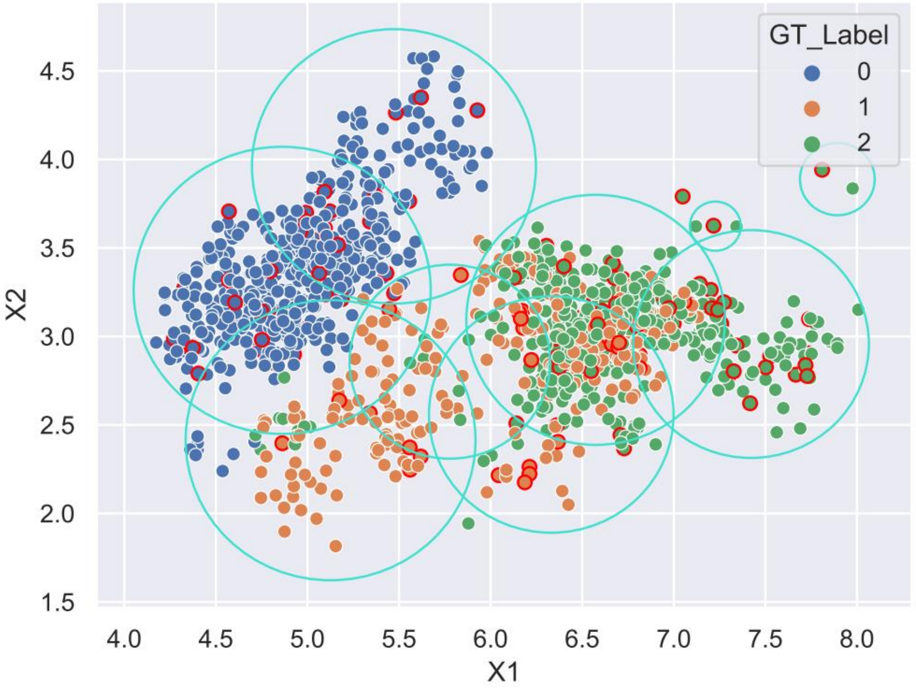
Approach – CluStream AL

- *CluStream*: Clustering Approach for Data Streams
  - Each cluster stored as Feature Vector: $\mathrm{C_i} = (LS_i^x, SS_i^x, LS_i^t, SS_i^t, n)$
  - Clustering evolves over time

- Idea: Extend the Feature Vectors with labeling statistics
  - **Change Detection:** Indicators for Concept Drifts can be derived
    - Cluster Radius
    - Cluster Density
    - Class Entropy inside cluster
  - **Locality of Changes** is considered
  - **Relevant Information** stored in cluster features
    - Cluster deleted when outdated

Approach – CluStream AL

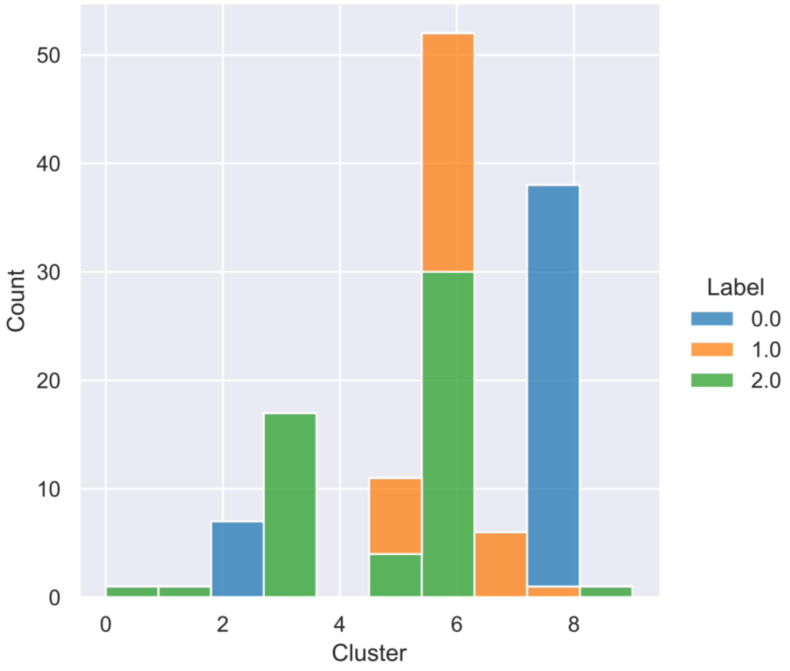
- **Clustering**



Experiment Setup:

- Data : Toy Data Set
- Based on Iris Data set
- CluStream
- Labeling Strategy: PAL

- **Label distributions per Cluster**



Approach – Open Questions

- **Cluster Statistics**
  - Which additional information should be stored in Feature Vectors?
  - How to derive a “Drift Score” for each cluster?
- **Active Learning Policy**
  - How to combine these statistics with an Active Learning Policy?
  - Distribute budget to cluster?
- **Model Structure**
  - Single Classifier or Ensemble?
  - One classifier for each cluster?
- **Adaption over Time**
  - Which information should remain
  - Incremental vs. Batch Learning

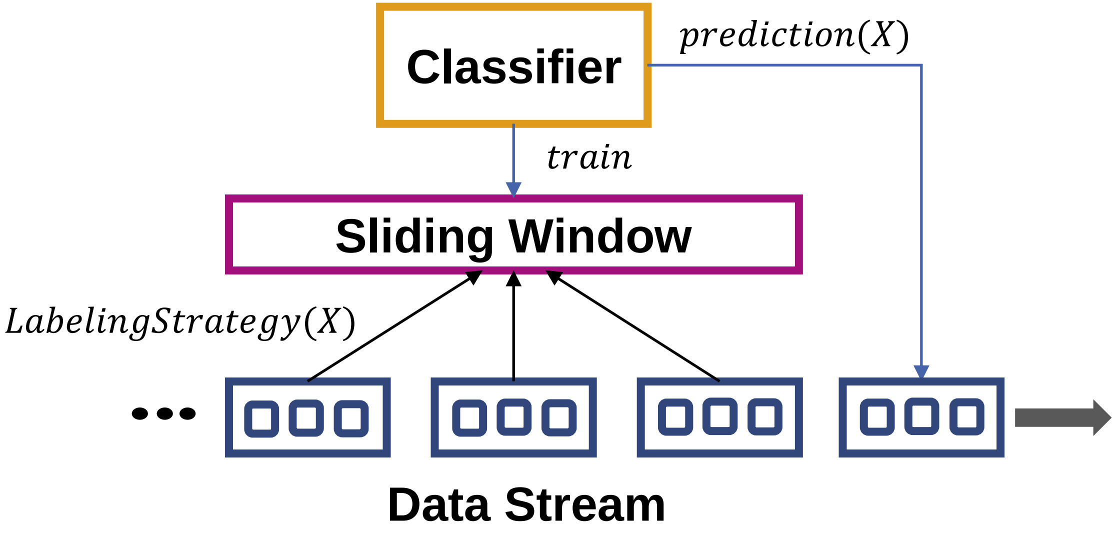
Appendix– Properties

- **Random based**: Label each instance with a certain probability
  - No historical data required
  - Labeled Instances uniformly distributed over entire Feature Space
- **Uncertainty based**: Label instances the classifier is least confident
  - E.g. margin posterior probability $\mathrm{margin}(X) = p(y_{c1}|X) - p(y_{c2}|X)$
  - No historical data required
  - Only labels around decision boundary
- **Local Density based:** Label most representative instances
  - Based on number of nearest neighbours
  - Historical data required

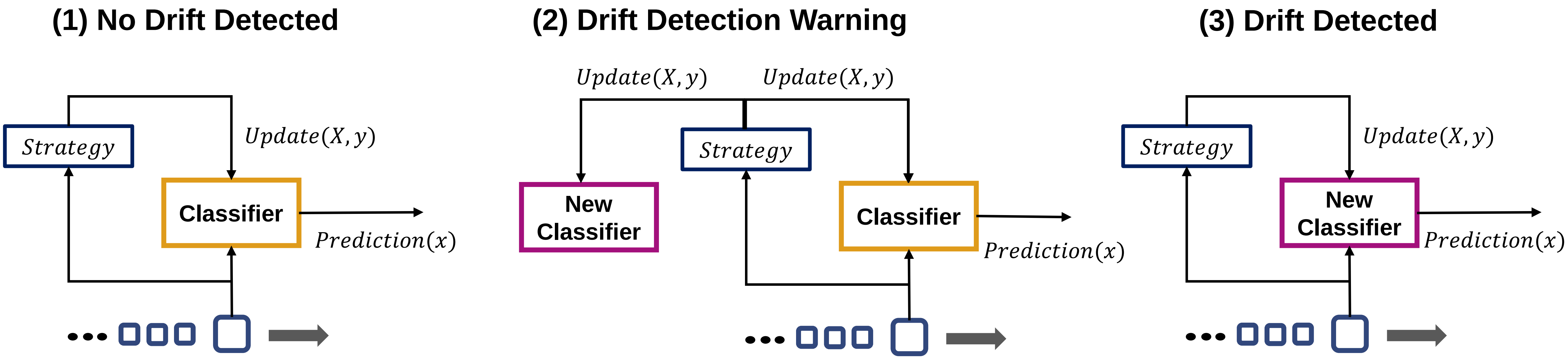
Appendix - Batch Incremental Methods

- **Window-Based Approach**
  - Uncertainty Sampling on Batch
  - Labeled instances in window
  - Classifier trained on window

- Problems:
  - Determine the training period
    - Overhead
    - Changes might get missed



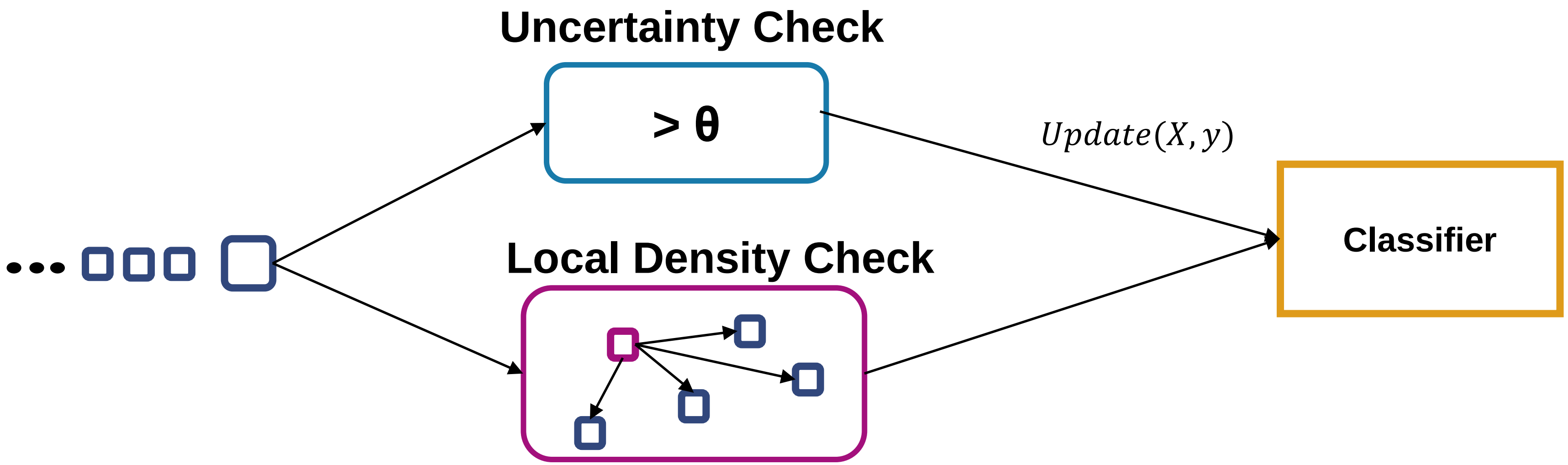
Appendix - Uncertainty Based AL



- Drift Detection based on Model accuracy or Label distribution
  - Slow detection in sparsely labeled streams

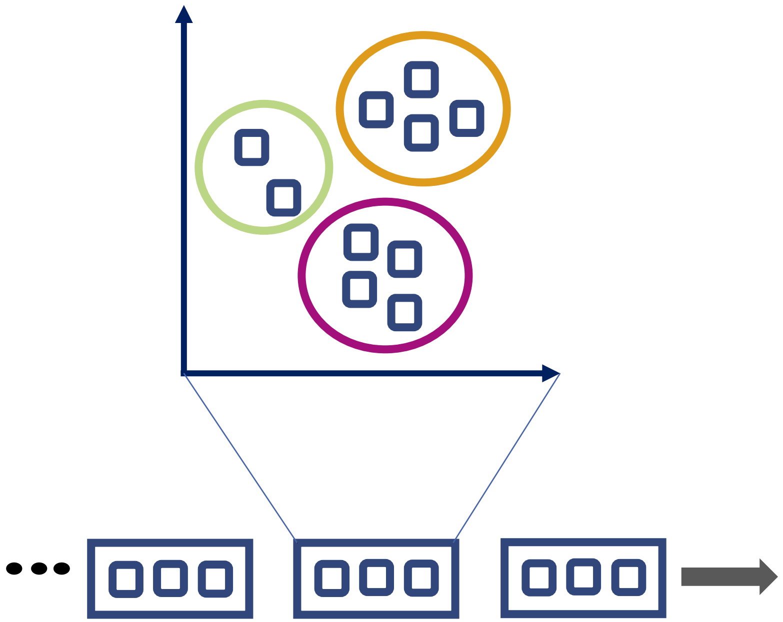
Appendix – Dual Query & Cognition Window

- Idea: use Uncertainty and Local Density Strategies
- Data is maintained in a specific Cognition Window
  - Based on time and Euclidian distances to each other
  - Used to calculate the local density



Appendix - Clustering Based

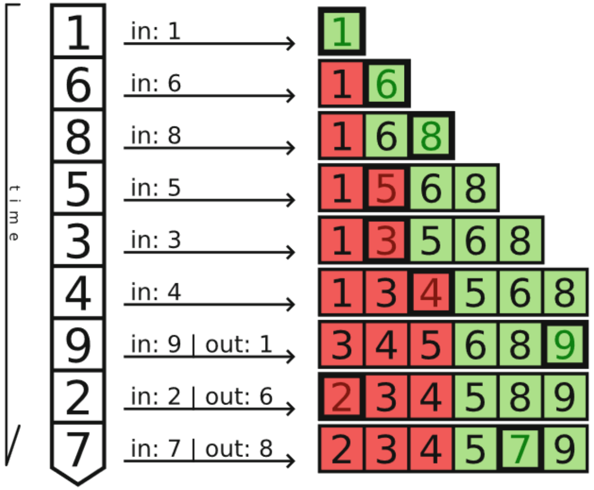
- Each incoming batch is classified and clustered
- Two-step-process to identify the instances to label
  1. **Macro Step:** Finding important clusters
     - Clusters ranked by class homogeneity metric
     - Further steps inside n best clusters
  2. **Micro Step:** Finding important instances
     - Selection based on two properties
       - Distance to centroid
       - Uncertainty of classifier
- Classifier is learned incrementally
- Representation property considered



Appendix - PAL

- **Uncertainty Based Sampling**
  - Label instances the classifier is least confident
    - E.g., margin posterior probability $\mathrm{margin}(X) = p(y_{c1}|X) - p(y_{c2}|X)$
    - $\mathbf{if}\ \ \mathrm{margin}(X) > \theta\ \ \mathbf{then}\ \ \mathrm{label}(X)$

- **Probabilistic Balanced Quantile Filter**
  - Instances accessed based on
    1. Spatial Usefulness
    2. Model Uncertainty
  - Threshold implemented as ordered list
    - Ordered by utility
    - Balancing Measurement to ensure Budget is met



Appendix – Time & Memory Requirements

- **Time Complexity** influenced by $Learning$ and $Labeling$

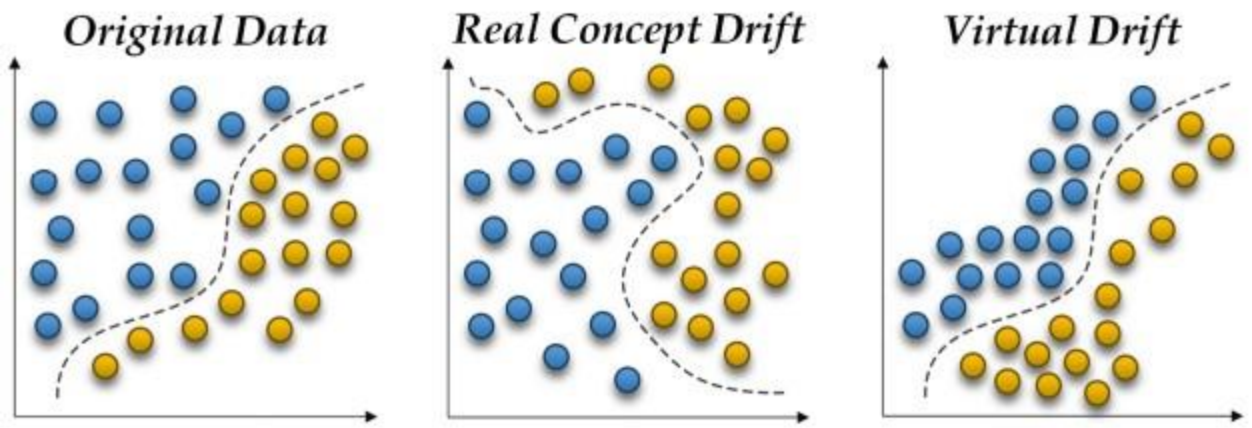
Time required for	Batch-Window	Basic-Incremental	DualQuery	ClusterBased
$Learning$	New Classifier each batch	Updated every instance	Updated every instance	Updated every instance
$Labeling$	Pool-Based sampling	Online sampling	2 * Online Sampling + Window calculations	Clustering + Pool-Based sampling

- **Memory requirements** influenced by
  - C : size of model structure
  - W: window size
  - B : Batch size

	Batch-Window	Basic-Incremental	DualQuery	ClusterBased
$Memory$	$O(C + W + B)$	$O(2 * C)$	$O(C + W)$	$O(C + Clustering + B)$

Handling Data Streams – Concept Drifts

- **Real Concept Drift**
  - Change in $p(y \mid X)$
  - Effects Decision Boundary
- **Virtual Concept Drift**
  - Only change in $p(X)$
  - No effects on Decision Boundary



Labeling Strategies – Labeling Budget

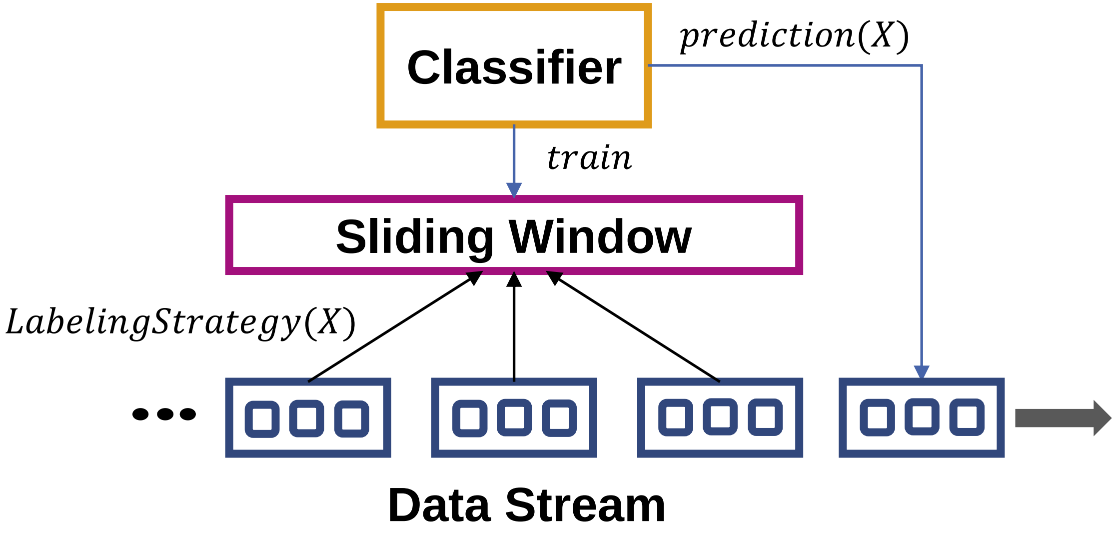
- Cost of labelling depends on application
  - Budget is defined a priori
  - Labeling only allowed if Budget not exceeded
- Problem: infinite sequence of data
  - Fixed Budget size for whole Stream unfeasible

- **Batch-incremental** setting
  - Fixed budget $B \in [0, 1]$ each Batch
  - Instances ranked by strategy
  - Label fraction B of best instances

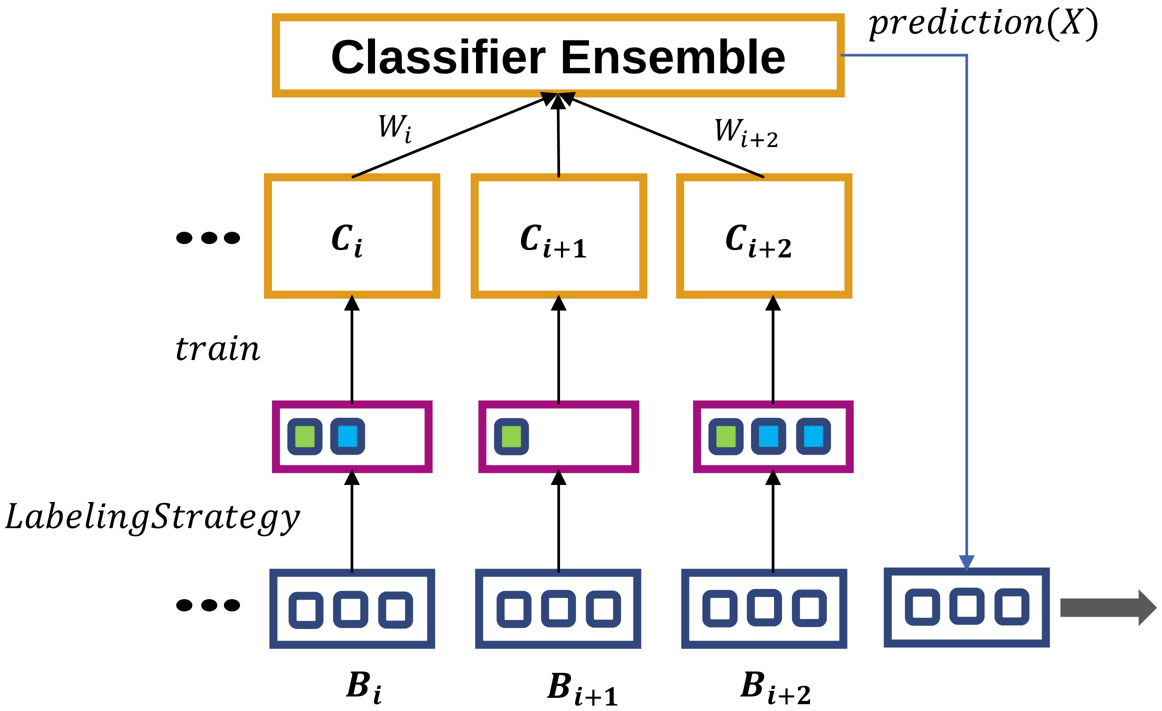
- **Instance-incremental** setting
  - Current budget spend $\hat{B} = \frac{\hat{v}}{w}$
    - w: Fixed size sliding window
    - $\hat{v}$: Number of labels in current window
  - Label allowed if $\hat{B} < B$

Approaches - Batch Incremental Methods

- **Window-Based Approach**
  - Uncertainty Sampling on Batch
  - Labeled instances in window
  - Classifier trained on window



- **Ensemble Approach**
  - Sampling on each Batch
  - Classifier for each Batch



Motivation – Data Stream Active Learning

- Data Stream $D = \ \dots, (X_{t-1}, y_{t-1}), (X_t, y_t), \dots$
  - $X_t \sim p_t(X)$: feature vector
  - $y_t \sim p_t(y)$: class label
  - (X_t, y_t) is sample from joint distribution $p_t(X, y)$
- Challenges in Data Stream Mining
  1. Volumes and arrival rates
  2. Memory constraints
  3. Real-time processing
  4. Changes in the data distribution (Concept Drifts)
- Label selection
  - Batch-processing: Pool-based techniques on stream batches
  - Instance-incremental: Decision on each instance at a time