Intelligent
Embedded Systems

Interactive Adaptive Learning

Daniel Kottke

Intelligent Embedded Systems, University of Kassel

July 8, 2018

What is “Interactive Adaptive Learning”?

Interactive Machine Leaning [9]
“We define iML-approaches as algorithms that can interact with both computational agents and human agents (in active learning: oracles) and can optimize their learning behavior through these interactions.”

Adaptive Stream Mining [2]
Adaptive Stream Mining deals “with time-changing data” which require “strategies for detecting and quantifying change, forgetting stale examples, and for model revision”.

What is “Interactive Adaptive Learning”?

We aim to bring different fields together...

- Interaction:
  Algorithms interact with both computational and human agents.
- Adaptation:
  The task probably changes over time and algorithms must adapt themselves.
- Learning:
  The agents optimize their behavior.

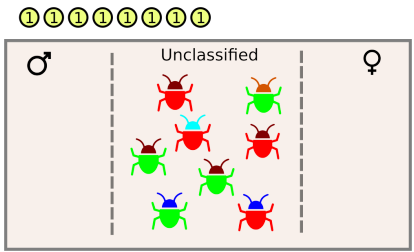
What is “Interactive Adaptive Learning”? – Examples

- Learning methods combining adaptive, active, semi-supervised, transfer, and reinforcement learning techniques
- Methods for big, evolving, or streaming data
- Methods for filtering, forgetting, and resampling of data
- Methods that detect change, outliers, frauds, or attacks
- Methods for timing the interaction and for combining different types of information of multi-modal data
- Cost-aware methods and methods for estimating the impact of employing additional resources, such as data or processing capacities, on the learning progress,
- Philiosophical, ethical, and legal questions

Game: Separatio

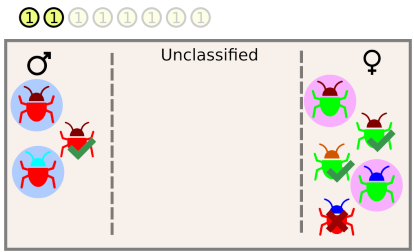
Thanks to Tuan Pham Minh and Ali Ahmed

Separatio Game



- Goal: Separate male and female bugs
- Init: *n* coins as budget
- Lab: Get sex of one bug for 1 coin
- Final: Sort every bug according to your classification hypothesis
- Eval: Every wrongly sorted bug costs 2 coins.

Separatio Game



- Goal: Separate male and female bugs
- Init: *n* coins as budget
- Lab: Get sex of one bug for 1 coin
- Final: Sort every bug according to your classification hypothesis
- Eval: Every wrongly sorted bug costs 2 coins.

Separatio Game

Features:

1. Antennas: yes, no
2. Color of head: blue, green, yellow
3. Color of dots: white, black
4. Number of dots: 1, . . . , 7

How to play?

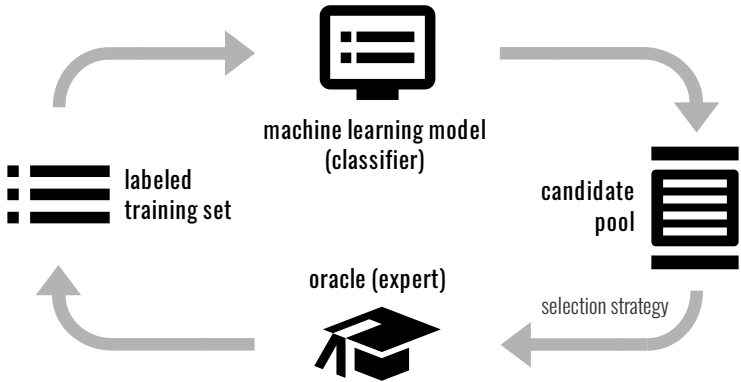
1. Google Play Store: Separatio
2. Our phones

Afterwards: Highscore and evaluation

Separatio – Evaluation

More information: Monday, 8 a.m. Machine Learning Session (Oceania IV)

Active Learning Cycle [14]



Challenges of Interactive Adaptive Learning

Challenges of Interactive Adaptive Learning

1. Finding an appropriate selection strategy
2. Deciding when to stop – Performance estimation
3. Active Learning with multiple, error-prone information sources
4. Multi-directional communication
5. Educating the expert, changing/influencing the environment (self-fulfilling prophecies
6. Extracting more information from humans
7. Evaluation and deployment in real-world applications

Agenda

1. Topic 1: Selection Strategies
2. Topic 2: Mining of Changing Streams
3. Topic 3: Managing Budgets of Stream-based Active Learning
4. Topic 4: Evaluation of Pool-based Active Learning
5. Application: Sorting Robot

Topic 1:
Selection Strategies

Active Learning by Settles [28]

- “Active learning systems attempt to overcome the labeling bottleneck by **asking queries** in the form of unlabeled instances to be labeled by an **oracle**. In this way, the active learner **aims to achieve high accuracy using as few labeled instances as possible**, thereby **minimizing the cost of obtaining labeled data**.” [28, p.5]

Definition of Active Learning

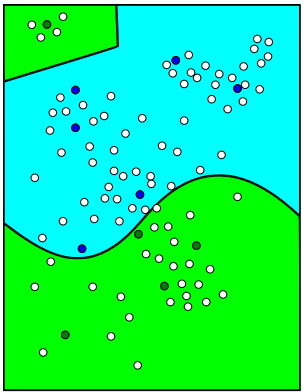
Definition:

- Active Component: ask queries to an oracle
- Improve the performance of a classifier
- Minimizing the cost of obtaining labeled data

Conclusion:

Active Learning optimizes a **performance** which is induced by a **classifier** through selecting the most beneficial **unlabeled instances** to be labeled by an **oracle** to build the **training basis**.

Visualization



What factors influence the decision?

- Density (improve the classifier, where decisions are important)
- Decision boundary (be specific, where change is expected)
- Label density (explore unexplored regions)

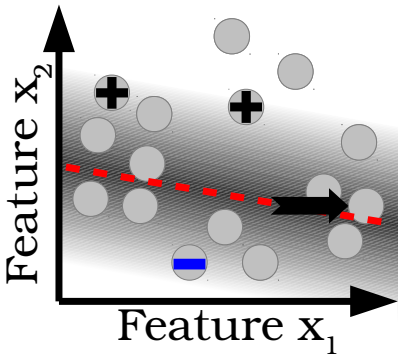
Influence Factors:

- **Decision boundary**: main criterion for decision making (prediction)
  - Proxy: posterior probability, margin, etc.
- **Reliability of decision**: identifies how sure one can be that the decision is already correct
  - Proxy: classifier ensemble diversity, labels distribution
- **Influence**: the influence of one instance for the complete dataset
  - Proxy: density, simulation
- **Class distribution**: are classes equally often represented
  - Proxy: class prior

Random Sampling

- Also called passive sampling
- Selects instances randomly for labeling
- Competitive approach
- Standard baseline
- Free of heuristics

Uncertainty Sampling [3]



Idea

Select those instances where we are least certain about the label

Approach:

- 3 labels preselected
- Linear classifier
- Use *distance to the decision boundary* as *uncertainty measure*

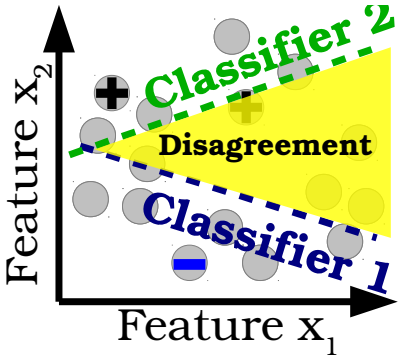
Discussion of Uncertainty Sampling

- ⊕ easy to implement
- ⊕ fast

- ⊖ no exploration (often combined with random sampling)
- ⊖ impact not considered (density weighted extensions exist)
- ⊖ problem with complex structures (performance can be even worse than random)

Influence factors: Decision boundary

Ensemble-Based Strategy [30]



Idea

Use disagreement between base classifiers

Approach

1. Get an initial set of labels
2. Split that set into (overlapping) subsets
3. On each subset, train a different base-classifier
4. Repeat until stop
5. On each unlabeled instance do
6. Apply all base-classifiers
7. Request label, if base-classifiers disagree
8. Update all base-classifiers
9. Go to step 4

Discussion of QbC

- ⊕ applicable to every classifier (even discriminative ones)
- ⊖ need more labels as some are hidden for some classifiers
- ⊖ training of multiple classifiers

Influence factors: Decision boundary, Reliability of decision

Expected Error Reduction [25]

- Simulates the acquisition of each label candidate and each possible outcome (class)
- Calculates the generalization error of the simulated new model
- Chooses the label with lowest generalization error

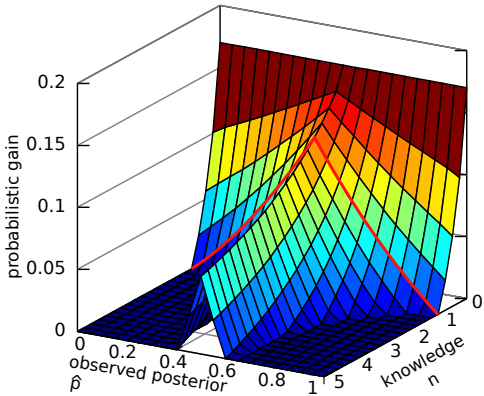
$$x^* = \operatorname{argmin}_x \sum_{i \in \{1,\dots,C\}} P_\theta(y_i \mid x) \left(\sum_{x' \in \mathcal{U}} 1 - P_{\theta^{+(x,y_i)}}(\hat{y} \mid x') \right)$$

Discussion of Expected Error Reduction

- ⊕ decision theoretic model
- ⊖ long execution time (closed form solutions for specific classifiers, approximations for speed up)

Influence factors: Decision boundary, Reliability of decision, Impact

Probabilistic Active Learning [19]



- Models the *true* posterior as being Beta-distributed
  - variance of posterior is correlated with the number of local observations
  - thereby omit the complex simulation of expected error reduction
- Calculates the performance improvement of the model

Discussion of Probabilistic Active Learning

- ⊕ decision theoretic model
- ⊕ fast w.r.t. expected error reduction

- ⊖ local number of labels required

Influence factors: Decision boundary, Reliability of decision, Impact

DUAL [5]

- combination of density weighted uncertainty sampling and standard (uniform) uncertainty sampling
- adaptive weights

Influence factors: Decision boundary, Impact

4DS [24]

- Uses four different scores for a classifier based on Gaussian mixtures (CMM):
  - distance, density, diversity, distribution
  - automatically weighted

Influence factors: Decision boundary, Class distribution, Impact, (Reliability of decision)

One-by-one vs. Batch Acquisition

Definition:

- One-by-one: subsequently selecting instances
- Batch: selects a specific number of labeling candidates for labeling at one time

Batch-Acquisition:

- Problem: most approaches would select very similar instances
- Approach: diversity score

Summary

- Uncertainty Sampling:
  selects instances near the decision boundary
- Query by Committee:
  minimizes classifier variance
- Expected Error Reduction:
  simulates acquisition of each candidate and each possible outcome
- Probabilistic Active Learning:
  calculates expected performance locally
- ... (there exist many methods)

Topic 2:
Mining of Changing Streams

Thanks to Georg Krempl and Vincent Lemaire

Motivation for Adaptive Interactive Stream Mining

Finance: High frequency trading

- Find correlations between the prices of stocks within the historical data
- Evaluate the stationarity of these correlations over the time
- Give more weight to recent data

Banking : Detection of frauds with credit cards

- Automatically monitor a large amount of transactions
- Detects patterns of events that indicate a likelihood of fraud
- Stop the processing and send an alert for a human adjudication

Medicine: Health monitoring

- Perform automatic medical analysis to reduce workload on nurses, . . .

Stream (Online) and Static (Offline) Learning

Big Data

- Static data
- Storage : distributed on several computers
- Query & Analysis : distributed and parallel processing
- Specific tools : Very Large Database (ex : Hadoop)

Fast Data

- Data in motion
- Storage: none (only buffer in memory)
- Query & Analysis: processing on the fly (and parallel)
- Specific Tools: CEP (Complex Event Processing)

Stream (Online) and Static (Offline) Learning

Issues in Timing and Availability of Supervision

- Feedback/Interaction might be limited
  e.g. costly labels due to limited time of domain expert
- Feedback is often delayed
  e.g. result of an experiment/investigation, or payment of a loan
- Even in applications with **big**/**fast** (e.g., unlabelled) data,
  some (e.g., labelled) data might be **sparse**/**delayed**!

Distinction: Online Learning vs. Online Deployment

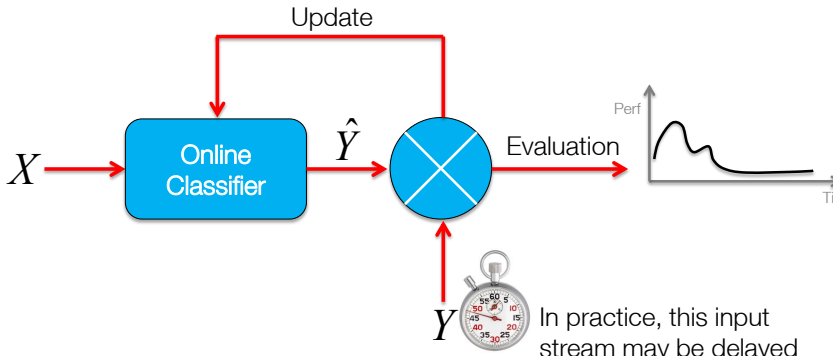
- Appropriateness depends on the practical application

Stream (Online) and Static (Offline) Learning

Particularities of Stream Classification

- Instances are received in subsets (one-by-one or in chunks)
- Instances might be discarded after being processed
- A hypothesis is produced after each instance is processed
  i.e. the system produces a series of hypotheses
- No distinct phases for learning and operation
  i.e. produced hypotheses can be used in classification
- Operates (often) as a real time system
- **Constraints: time, memory, . . .**
- **i. i. d. assumption does not hold!**
- Neither prediction nor learning ever stops

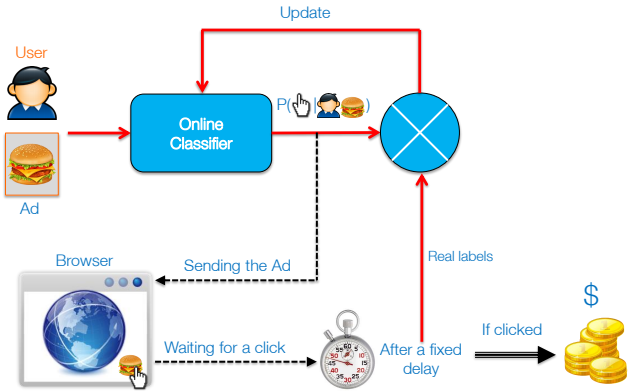
Adaptive Stream Classification: Implementation



A on-line classifier predicts the class label of tuples **before** receiving the true label …

Adaptive Stream Classification Application Example

Online Advertising Targeting



Adaptive Stream Classification: Summarizing the Main Challenges [1]

Volume and **Velocity**:

- processing high volumes of data in limited time
- no inital data, but possibly infinite (unknown) length of stream

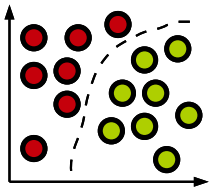
Volatility:

- dynamic environment with ever-changing patterns
- old data might become useless or even misleading due to **change**

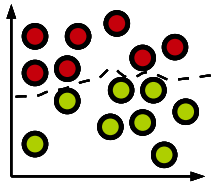
[1] See e.g., [6, 20, 7].

Types of Change – Change might affect

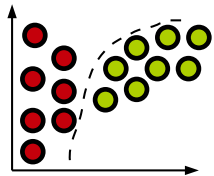
- the target, e.g. the variable Y changes
- the available features X, e.g. new are added or old ones removed
- the distributions, so called concept (or population) drift or shift [27, 12, 22]



Original distribution
$P(X, Y)$



Real Concept Drift:
$P(Y|X)$ has changed

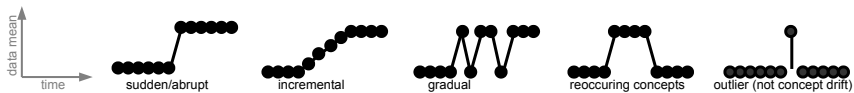


Virtual Concept Drift:
$P(Y|X)$ is static

[1] Illustrations: [41]

Types of Drift

- By the affected distributions: E.g. $P(X, Y)$, $P(X)$, $P(Y)$, $P(Y|X)$, $P(X|Y)$
- Smoothness of concept transition: sudden shift vs. gradual drift
- Singular or recurring contexts: with recurring context, obsolete data and models gain relevance again
- Systematic or unsystematic: E.g. distributions change according to patterns
- Real or virtual: change affects the decision boundary or solely the feature distribution (or noise)



[1] See e.g., [38, 8, 20, 33].
[1] Illustration from [41]

Hoeffding Trees

Very Fast Decision Trees

Problem

- Massive amount of data
- Considering every instance for every node ?

Basic Idea

Very Fast Decision Tree (VFDT), suggested by Domingos and Hulten in 2000 [4]:

- Calculate quality measure for each attribute (e.g. entropy)
- Decide with **Hoeffding bound** if enough data exists to select split attribute
- If enough data exists, add split to tree and create subnodes, start learning at each subnode;
  Otherwise wait for more data

Hoeffding Bound

Also denoted as additive Chernoff bound (Hoeffding, 1963)

- Given:
  - Real-valued random-variable r with range R, *arbitrarily distributed*
  - User specified confidence $1 - \delta$
  - True mean $\bar{r}_0$ of r is unobservable,
    but sample mean $\bar{r}$ can be calculated
- After n independent observations of R, test whether

$$\bar{r}_0 \geq \bar{r} - \epsilon$$

with

$$\epsilon = \sqrt{\frac{R^2 \log(1/\delta)}{2n}}$$

Very Fast Decision Trees – Basic Idea (cont'd)

Very Fast Decision Tree (VFDT), suggested by Domingos and Hulten in 2000 [4]:

- Calculate quality measure for each attribute (e.g. entropy)
- Decide with **Hoeffding bound** if enough data exists to select split attribute
- If enough data exists, add split to tree and create subnodes, start learning at each subnode; Otherwise wait for more data

- Let X_a and X_b the first and second best attributes (w.r.t. a heuristic measure)
- Let $\bar{G}(X_i)$ be the heuristic measure to chose split attributes, such that the bigger $\bar{G}$, the better (e.g. information gain)
- Apply Hoeffding bound to

$$\Delta\bar{G} = \bar{G}(X_a) - \bar{G}(X_b)$$

- If $\Delta\bar{G} > \epsilon$, we are confident that difference between X_a and X_b is larger zero Thus, choose X_a for split

Very Fast Decision Trees – Some remarks

- Instance is passed to a leaf,
  and used only for deciding upon additional split there
- Only counts are kept and updated
- Time complexity of processing a new instance is $O(ldvc)$ with
  l maximum tree depth, d number of attributes,
  v max. number of values per attribute, c number of classes
- The time needed to process a new instance does not depend on the number of previously seen instances
  A single-pass algorithm, *usable on fast streams!*
- Considers *all* observations, *no forgetting*!
  If concept changes, many more instances of a new concept are needed to outweight instances of old concept(s)
  Only applicable on streams with static concepts, not on drifting concepts!

Concept-Adapting Very Fast Decision Trees

Background

- Include forgetting, avoid multiple passes over data
- Concept-adapting VFDT: Hulten, Spencer, and Domingos 2001 [10]

Concept-Adapting Very Fast Decision Trees

Basic Idea

- Recompute quality measures at *every* node (every fixed number of new observations) within a window
- If another split attribute yields similar performance, learn alternative subtree for this split attribute. Thus, we have a set of alternative subtrees for every node (least promising ones are dismissed if memory is getting low)
- If accuracy of a new subtree is significantly lower than existing one, dismiss the alternate subtree
- If accuracy of new subtree is significantly higher, exchange subtrees
- **Note**: Alternate subtrees are learnt with new instances only, thus replacement of old subtree yields forgetting

Adaptive Stream Learner

Categorization of Adaptive Stream Classifier Technologies[2]

Memory

- **Forgetting Mechanism**
  - Abrupt Forgetting: instances are either inside or outside the training window, based on their age or their order [1]
  - Sampling: instances are selected according to some probability
    E.g. Reservoir Sampling [32]
  - Gradual Forgetting: instances' weights decrease with their age (full memory approach!)
    E.g. linear [16], exponential [13]

[2] See e.g., [7] (partially).

Categorization of Adaptive Stream Classifier Technologies[3]

Learning

- **Learning Mode**
  - Incremental (by updating an existing model, CVFDT [10])
  - Retraining (a new model from scratch, requires more buffered data)
- **Adaptation Methods**
  - Blind (without explicit change detection) vs. Informed
    (adaptation is triggered by e.g. a change detector like in CVFDT or by recognizing a context like in [34])
  - Global vs. Local Replacement (i.e. the whole model is replaced, or only parts of it)
  - Single Model vs. Ensemble

[3] See e.g., [7] (partially).

Common Assumption:

Information (features, labels) on each instance is

- *correct* (i.e. reliable),
- *complete* (i.e. true labels and features finally known),
- *immediately available* (i.e. before the *next* instance must be processed)
- available at *no cost* and *without control* by the classifier *on label selection*.

Summary and Concluding Remarks

Summary

- Data Stream Challenges:
  - Volume and Velocity: low time & space complexity required
  - Volatility: change, e.g. concept drift that requires adaptation
- Variety of approaches, categorized by
  - data management and forgetting mechanisms (e.g. sliding windows)
  - learning mode (e.g. incremental) and adaptation methods (e.g. actively upon change detection)
- Applications often present data in multiple streams:
  e.g. features and labels arrive at different times

Summary and Concluding Remarks

(Some) Open Challenges

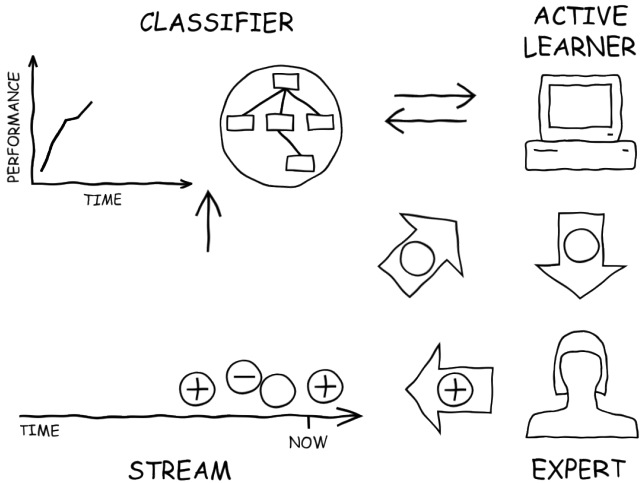
- Imbalanced Classes
- Sparse Labels: Semi-Supervised Learning
- Costly Labels: Active Learning
- Delayed Labels: Temporal Transfer Learning

Literature Surveys

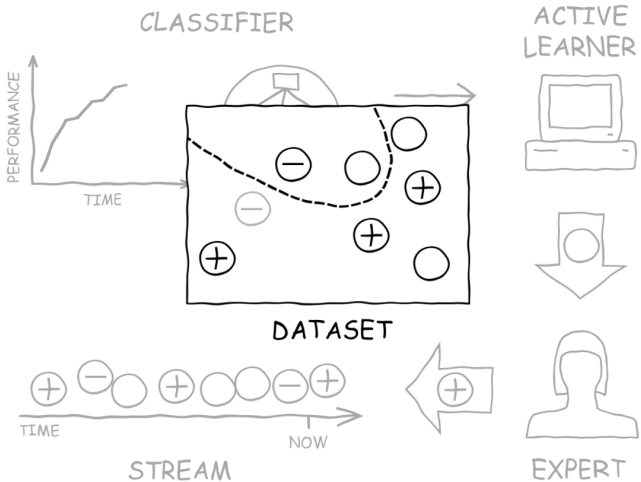
- Overview & Taxonomy of Techniques: [7]
- Open Challenges: [20]
- Applications: [41]
- Ensembles: [17]

Topic 3:
Managing Budgets of Stream-based Active Learning

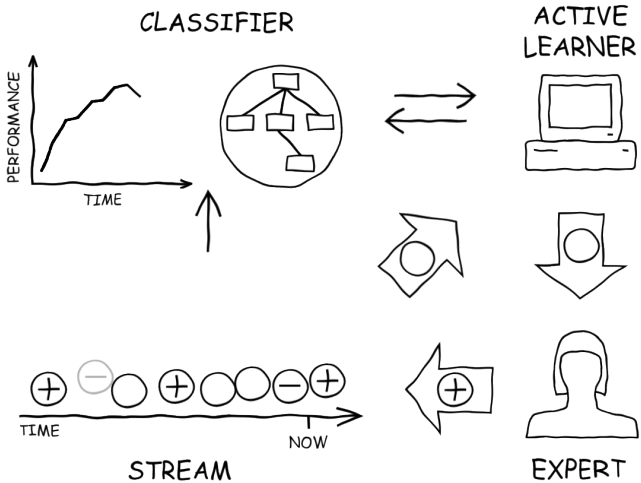
Introduction



Introduction



Introduction



Challenges in Stream Active Learning

Challenges of Stream Active Learning

Pool Active Learning

- Where to buy instances (spatial usefulness)?
  - Balance Exploration and Exploitation in the dataspace

Stream Active Learning

- Where to buy labels (**spatial usefulness**)?
- **Consider Drift**
  - Labels might change over time and have to be validated
  - Lifetime of labels
- When to buy labels (**temporal usefulness**)?
  - Balance Exploration and Exploitation in time

Spatial Usefulness

Where to buy labels?

- Use scores from pool-based methods like
  - Uncertainty sampling [11, 31, 35, 40]
  - Query by committee [26, 37]
  - Probabilistic active learning [15]

Approach

Find best instances spatially (based on feature vectors) balancing:

- exploration (observe unsampled regions)
- exploitation (acquire labels in regions near decision boundaries to elaborate the decision)

Consider Drift

Motivation

Labels might change over time and have to be validated

- Drift can affect *any region* of feature space [39]

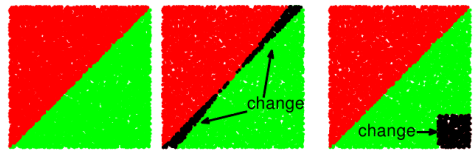
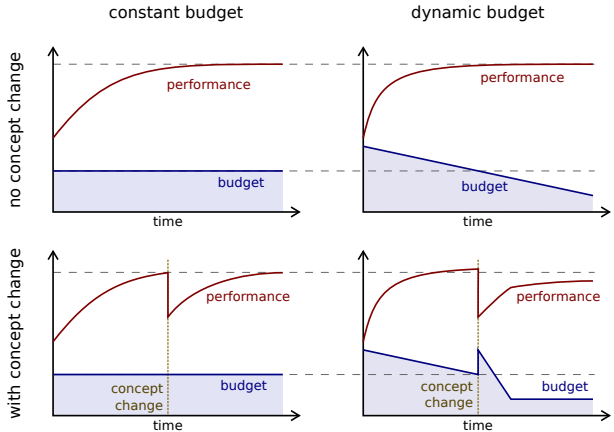


Image from [39], Figure 6, page 605.

Budget in Streams

- Pools: absolute number (e.g. stop after 40 labels)
- Streams: relative definition necessary (e.g. buy 10%)
- How to distribute the budget over time?
  - constantly (every 10th label $\rightarrow$ no spatial selection necessary)
  - almost constantly (with a small tolerance window) [15]
  - bounded (budget should not exceed 10%) [40]
  - dynamic (budget changes over time)

Temporal usefulness (When to buy?)



Temporal usefulness (When to buy labels?)

- Labels in the beginning are more beneficial as they affect more future decisions (resp. after changes)
- But: one does not know when change take place
- Standard technique: constant budget

Exploration vs. Exploitation

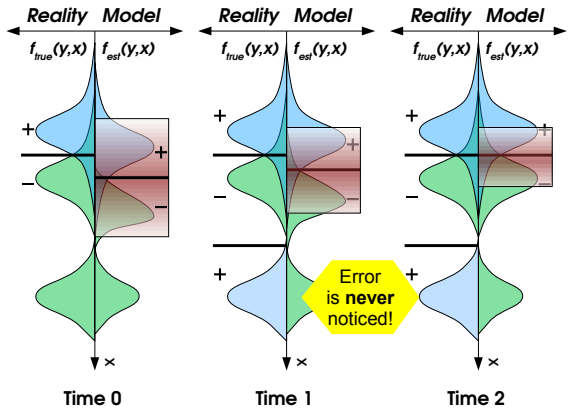
- Exploration: Sample randomly to be able to detect change
- Exploitation: Sample the most promising labels
- How to cope with gradual drifts?
- High budgets after change might cause problems due to less spatial usefulness

Example: Self Lock-In Problem (for US)

Motivation

Why not simply apply
active learning strategies
from static (*iid*) streams?

- Example:
  Uncertainty sampling,
  drifting distributions
- Error is *never* even noticed!
- **Active learner (self) lock-in**
  on an outdated hypothesis
- **Caveat:**
  Drift can occur anywhere in the
  feature space, as noted by [40]
- **Remedy:** Sampling from the
  whole feature space.



Temporal Usefulness

Batch/Chunk-Based Processing [11, 18]

Define chunk size w:

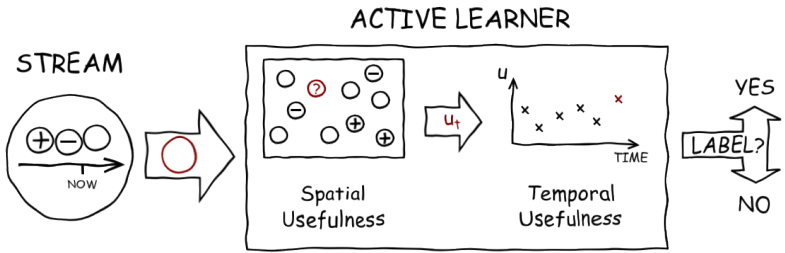
1. Collect w instances from the stream into a chunk
2. Select instances with pool active learning according to budget
3. Train Classifier
4. Repeat

Discussion

- $\oplus$ Easy to understand/implement
- $\ominus$ Delays training to the end of the batch

One-by-one Processing [15]

- Determines usefulness of one instance when it arrives
- Threshold balances acquisition



One-by-one Processing [15, 40]

- Determines usefulness of one instance when it arrives
- Threshold balances acquisition

Discussion

⊕ Training can be processed immediately

⊖ Needs additional budgeting component

▸ to chunk

Temporal Usefulness

Zliobaite et al. [40]

- Spatial selection: uncertainty sampling (exploitation) with random sampling (exploration)
- Temporal selection: adaptive threshold (ensures that budget is not exceeded)

Kottke et al. [15]

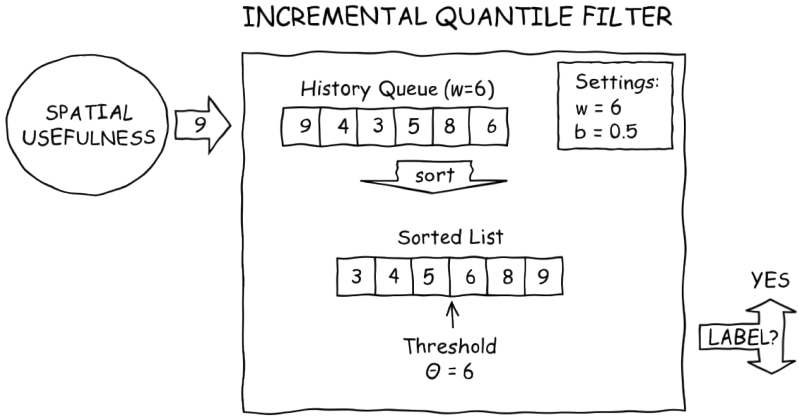
- Spatial selection: Probabilistic active learning
- Temporal selection: balanced incremental quantile filter (BIQF) (ensures that budget is within a given tolerance window)

Adaptive Threshold [40]

init $\theta = 1$, $s \in (0, 1]$

1. **if** budget not exceeded (approx.):
2. **if** $P(y^* \mid x) < \theta$:
3. $\theta \leftarrow \theta(1 - s)$
4. get label
5. **else**:
6. $\theta \leftarrow \theta(1 + s)$
7. do not get label

Incremental Quantile Filter [15]



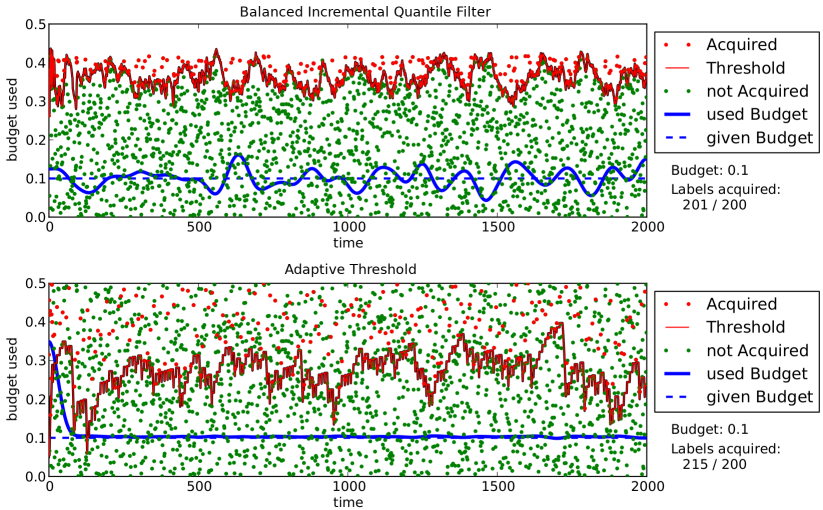
Balancing [15]

- Tolerance window (w_{tol}): maximal difference of acquisitions between current and the target budget
- Idea: If there are label acquisitions left decrease threshold θ (and vice versa)

$$\theta_{\text{bal}} = \theta - \Delta \cdot \frac{acq_{\text{left}}}{w_{\text{tol}}}$$

θ - original threshold
θ_{bal} - balanced threshold
Δ - Data range of IQF window
w_{tol} - Tolerance window size

Discussion


Balanced Incremental Quantile Filter
budget used
time
Acquired
Threshold
not Acquired
used Budget
given Budget
Budget: 0.1
Labels acquired:
201 / 200
Adaptive Threshold
budget used
time
Acquired
Threshold
not Acquired
used Budget
given Budget
Budget: 0.1
Labels acquired:
215 / 200

Topic 4:
Evaluation of Pool-based Active Learning

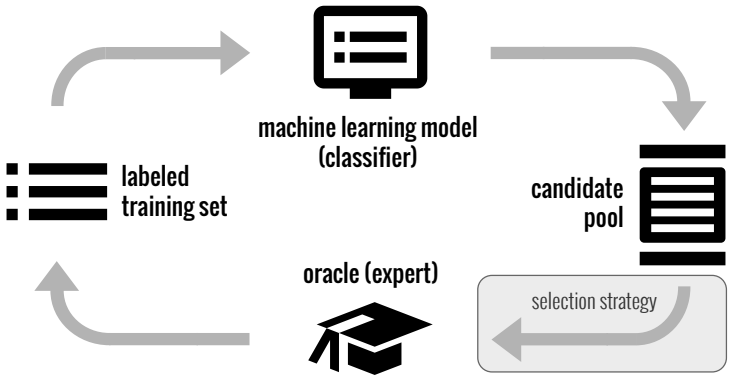
Motivation[4]

The evaluation methodology should be

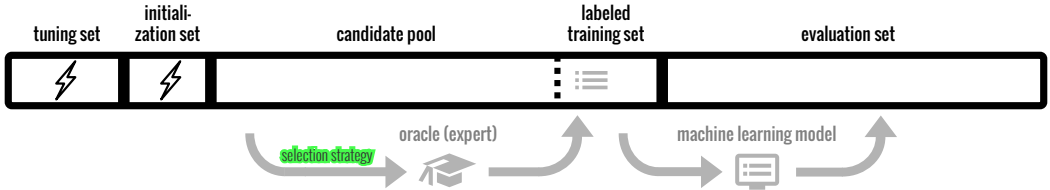
1. reliable
   - robust to varying seeds or shuffling data
   - reproducible (well-described, availability of data)
2. realistic
   - valid assumptions for real applications
3. comparable
   - development of a standardized active learning evaluation gold standard to compare algorithms without reimplementing

[4] Based on [14]

Recap: Active Learning Cycle [28]



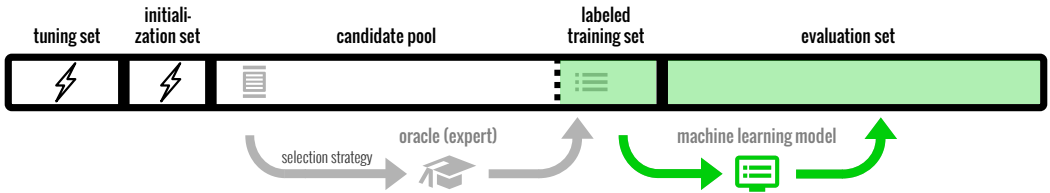
A Different View on the Active Learning Cycle



We want to evaluate the performance of the **selection strategy**.

Reliable evaluation

Evaluating the Model's Performance



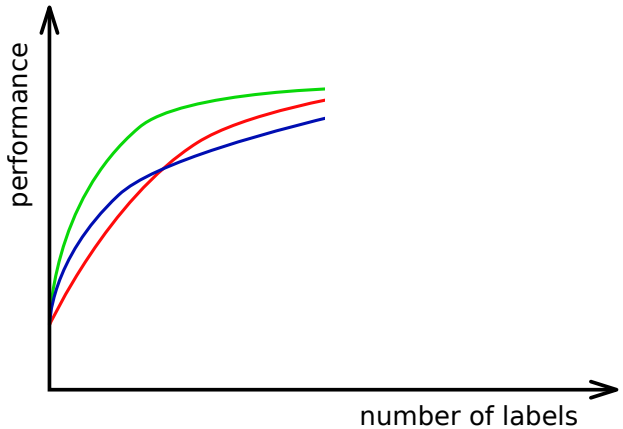
- training set is subsequently filled with selected candidates
- the learned model is evaluated on a hold-out evaluation set

Which performance measure should be used?

- depends on the application
  - balanced class priors (e.g., accuracy, error)
  - unbalanced class priors (e.g., f1-score, AUROC)
- complexity [21]:
  - point measures (e.g., accuracy, precision, recall)
  - integrated measures (e.g., AUROC, H-Measure)

How to interpret the results of a learning curve?

- converging as fast as possible
- converging to the highest overall value



How to summarize results from a learning curve?

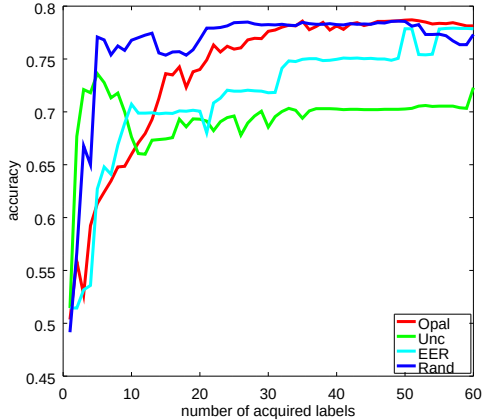
- Table at specific time points (early, mid, late)
- Area under the learning curve, mean (depends on stopping point)
- deficiency [36]
- data utilization rate [23]

How to evaluate statistical significance?

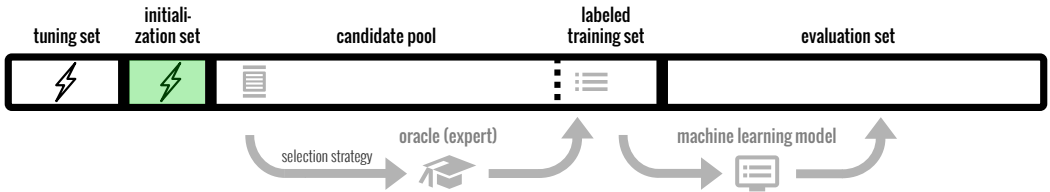
- Which values to compare?
  - **not** across label acquisitions (highly correlated) but across multiple repetitions
  - at which point in time?
- Statistical tests
  - t-Test cmp. mean (assumes that mean is normal distributed)
  - Wilcoxon Signed Rank Test cmp. tendency (parameter-free test)
- always present results with **statistical significance** and **effect size**

How many repetitions are required?

Comparison of algorithms using 5-fold cross validation

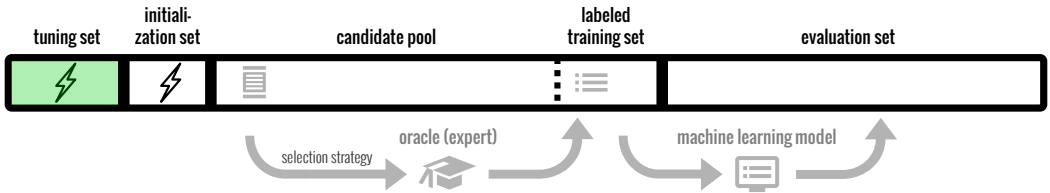


Initialization of Instance Selection



- Cannot be class-specific, as labels are unknown
- Often random (How to tune the number of random samples?)

Parameter Tuning



1. Determine hyperparameter and fix them across selection methods
2. How to tune without labels?

Parameter Tuning

- tuning instances should be considered in the number of acquisitions
- how many instances should be used for tuning? (many classifiers are sensitive to the number of instances)
- normally, no instances for supervised parameter tuning available
- tuning parallel to sampling may be complicated

Realistic evaluation

Real applications oft are more challenging

- Often highly specialized (hard to transfer approaches to related domains)
- Imperfect labelers (experts might be wrong)
- In real-world only one shot (mean results are not representative)
- Labels are not always available (in time and space)
- Performance guarantees (cmp. random sampling)
- Assess online performance of an actively trained classifier
- Different costs for different annotations or classes
- Ground truth might not be available

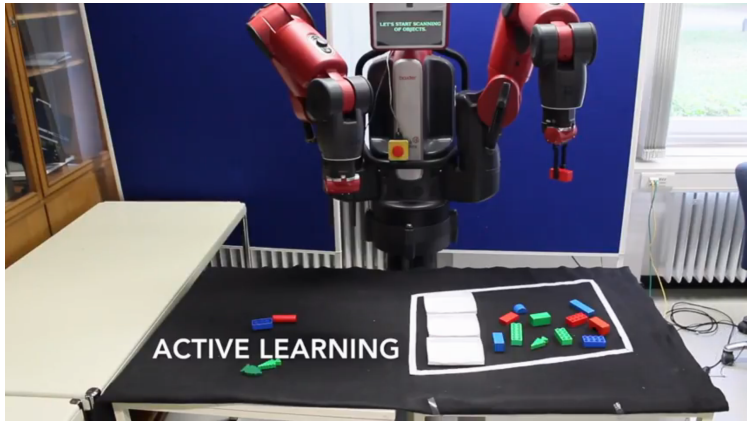
Comparable evaluation

Discussion on an Evaluation Gold Standard

- Use exactly the same robust classifier for every AL method when comparing and try to sync the parameters of these classifiers.
- Capture the effect of different AL methods on multiple datasets using at least 50 repetitions.
- Start with an initially unlabeled set. If you need initial training instances, sample randomly and explain when to stop.
- Use either a clear defined stopping criterion or enough label acquisitions (sample until convergence).
- Show learning curves (incl. quartiles) with reasonable performance measures.
- Present pairwise differences in terms of significance and effect size (Wilcoxon signed rank test).

Application:
Sorting Robot

Application: Sorting Robot



https://youtu.be/TMd4VBBuTt0

Thanks

- Adrian Calma (University Kassel) – co-organization
- Robi Polikar (Rowan University) – co-organization
- Georg Krempl (University Utrecht) – slides
- Vincent Lemaire (Orange Labs) – slides
- Tuan Pham Minh (University Kassel) – bug app
- Ali Ahmed (University Kassel) – bug app
- Marek Herde (University Kassel) – bug app

Workshop on IAL @ ECMLPKDD (Dublin)

Bibliography I

[1] Brian Babcock, Shivnath Babu, Mayur Datar, Rajeev Motwani, and Jennifer Widom.
Models and issues in data stream systems.
In *Proceedings of the Twenty-first ACM SIGMOD-SIGACT-SIGART Symposium on Principles of Database Systems*, PODS '02, pages 1–16, New York, NY, USA, 2002. ACM.

[2] Albert Bifet.
Adaptive stream mining: Pattern learning and mining from evolving data streams.
In *Proceedings of the 2010 Conference on Adaptive Stream Mining: Pattern Learning and Mining from Evolving Data Streams*, pages 1–212, Amsterdam, The Netherlands, The Netherlands, 2010. IOS Press.

[3] David Cohn, L. Atlas, R. Ladner, M. El-Sharkawi, R. II Marks, M. Aggoune, and D. Park.
Training connectionist networks with queries and selective sampling.
In *Advances in Neural Information Processing Systems (NIPS)*. Morgan Kaufmann, 1990.

[4] Pedro Domingos and Geoff Hulten.
Mining high-speed data streams.
In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining (KDD00)*, KDD '00, pages 71–80. ACM, 2000.

Bibliography II

[5] Pinar Donmez, JaimeG. Carbonell, and PaulN. Bennett.
Dual strategy active learning.
In *Machine Learning: ECML 2007*, volume 4701 of *Lecture Notes in Computer Science*, pages 116–127. Springer Berlin Heidelberg, 2007.

[6] Wei Fan and Albert Bifet.
Mining big data: Current status, and forecast to the future.
SIGKDD Explor. Newsl., 14(2):1–5, April 2013.

[7] João Gama, Indrė Zliobaitė, Albert Bifet, Mykola Pechenizkiy, and Abdelhamid Bouchachia.
A survey on concept drift adaptation.
ACM Computing Surveys, 46(4):1–44, 2014.

[8] Vera Hofer and Georg Krempl.
Drift mining in data: A framework for addressing drift in classification.
Computational Statistics and Data Analysis, 57(1):377–391, 2013.

[9] Andreas Holzinger.
Interactive machine learning (iml).
Informatik Spektrum, 39(1), 2016.

Bibliography III

[10] Geoff Hulten, Laurie Spencer, and Pedro Domingos.
Mining time-changing data streams.
In *KDD '01: Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 97–106, New York, NY, USA, 2001. ACM.

[11] Dino Ienco, Albert Bifet, Indrė Zliobaitė, and Bernhard Pfahringer.
Clustering based active learning for evolving data streams.
In Johannes Fürnkranz, Eyke Hüllermeier, and Tomoyuki Higuchi, editors, *Proceedings of the 16th Int. Conf. on Discovery Science (DS), Singapore*, volume 8140 of *Lecture Notes in Artificial Intelligence*, pages 79–93. Springer, 2013.

[12] Mark G. Kelly, David J. Hand, and Niall M. Adams.
The impact of changing populations on classifier performance.
In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 367–371, 1999.

[13] Ralf Klinkenberg.
Learning drifting concepts: Example selection vs. example weighting.
Intell. Data Anal., 8(3):281–300, August 2004.

Bibliography IV

[14] Daniel Kottke, Adrian Calma, Denis Huseljic, Georg Krempl, and Bernhard Sick.
Challenges of reliable, realistic and comparable active learning evaluation.
In *Interactive Adaptive Learning Workshop @ ECMLPKDD 2017*, CEUR Workshop Proc. 1924, pages 2–14, 2017.

[15] Daniel Kottke, Georg Krempl, and Myra Spiliopoulou.
Probabilistic active learning in data streams.
In Tijl De Bie and Elisa Fromont, editors, *Advances in Intelligent Data Analysis XIV - 14th Int. Symposium, IDA 2015, St. Etienne, France*, volume to appear of *Lecture Notes in Computer Science*. Springer, 2015.

[16] Ivan Koychev.
Gradual forgetting for adaptation to concept drift.
In *In Proceedings of ECAI 2000 Workshop Current Issues in Spatio-Temporal Reasoning*, pages 101–106, 2000.

[17] Bartosz Krawczyk, Leandro L. Minku, João Gama, Jerzy Stefanowski, and Michał Woźniak.
Ensemble learning for data stream analysis: A survey.
Information Fusion, 37:132 – 156, 2017.

Bibliography V

[18] Georg Krempl, Tuan Cuong Ha, and Myra Spiliopoulou.
Clustering-based optimised probabilistic active learning (copal).
In Nathalie Japkowicz, Stan Matwin, Nathalie Japkowicz, and Stan Matwin, editors, *Proc. of the 18th Int. Conf. on Discovery Science (DS 2015)*, volume 9356 of *Lecture Notes in Computer Science*, pages 101–115. Springer, 2015.

[19] Georg Krempl, Daniel Kottke, and Vincent Lemaire.
Optimised probabilistic active learning (OPAL) for fast, non-myopic, cost-sensitive active classification.
Machine Learning, Special Issue of ECML PKDD 2015, 2015.

[20] Georg Krempl, Indrė Zliobaitė, Dariusz Brzeziński, Eyke Hüllermeier, Mark Last, Vincent Lemaire, Tino Noack, Ammar Shaker, Sonja Sievi, Myra Spiliopoulou, and Jerzy Stefanowski.
Open challenges for data stream mining research.
SIGKDD Explorations, 16(1):1–10, 2014.
Special Issue on Big Data.

[21] Charles Parker.
An analysis of performance measures for binary classifiers.
In *Proceedings of the 11th IEEE International Conference on Data Mining (ICDM2011)*, pages 517 – 526. IEEE, 2011.

Bibliography VI

[22] Joaquin Quiñonero-Candela, Masashi Sugiyama, Anton Schwaighofer, and Neil D. Lawrence, editors.
Dataset Shift in Machine Learning.
MIT Press, 2009.

[23] T Reitmaier and B Sick.
Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4DS.
In *Information Sciences - Informatics and Computer Science Intelligent Systems Applications*, volume 230, pages 106–131, 2013.

[24] Tobias Reitmaier and Bernhard Sick.
Let us know your decision: Pool-based active training of a generative classifier with the selection strategy 4ds.
Information Sciences, 230:106–131, 2013.

[25] Nicholas Roy and Andrew McCallum.
Toward optimal active learning through sampling estimation of error reduction.
In *Proc. of the 18th Int. Conf. on Machine Learning, ICML 2001, Williamstown, MA, USA*, pages 441–448, San Francisco, CA, USA, 2001. Morgan Kaufmann.

Bibliography VII

[26] Joung Woo Ryu, Mehmed M Kantardzic, Myung-Won Kim, and A Ra Khil.
An efficient method of building an ensemble of classifiers in streaming data.
In *Big Data Analytics*, pages 122–133. Springer, 2012.

[27] Jeffrey C. Schlimmer and Richard H. Granger.
Beyond incremental processing: Tracking concept drift.
In *AAAI*, pages 502–507, 1986.

[28] Burr Settles.
Active learning literature survey.
Computer Sciences Technical Report 1648, University of Wisconsin-Madison, Madison, Wisconsin, USA, 2009.

[29] Burr Settles.
Active Learning.
Number 18 in Synthesis Lectures on Artificial Intelligence and Machine Learning. Morgan and Claypool Publishers, 2012.

[30] H. Sebastian Seung, Manfred Opper, and Haim Sompolinsky.
Query by committee.
In Warmuth M.K. and Valiant L.G., editors, *Proc. of the fifth workshop on computational learning theory*. Morgan Kaufmann, 1992.

Bibliography VIII

[31] Jasmina Smailović, Miha Grčar, Nada Lavrač, and Martin Žnidaršič.
Stream-based active learning for sentiment analysis in the financial domain.
Information Sciences, 285:181–203, 2014.

[32] Jeffrey S Vitter.
Random sampling with a reservoir.
ACM Transactions on Mathematical Software (TOMS), 11(1):37–57, 1985.

[33] Geoffrey I. Webb, Loong Kuan Lee, François Petitjean, and Bart Goethals.
Understanding concept drift.
CoRR, abs/1704.00362, 2017.

[34] Gerhard Widmer and Miroslav Kubat.
Learning in the presence of concept drift and hidden context.
Machine Learning, 2369–101, 1996.

[35] Michał Woźniak, Bogusław Cyganek, Andrzej Kasprzak, Paweł Ksieniewicz, and Krzysztof Walkowiak.
Active learning classifier for streaming data.
In Francisco Martínez-Álvarez, Alicia Troncoso, Héctor Quintián, and Emilio Corchado, editors, *Proc. of the 11th Int. Conf. on Hybrid Artificial Intelligent Systems*, pages 186–197. Springer International Publishing, 2016.

Bibliography IX

[36] Erelcan Yanik and Tevfik Metin Sezgin.
Active learning for sketch recognition.
Computers and Graphics (Pergamon), 52:93–105, 2015.

[37] Xingquan Zhu, Peng Zhang, Xiaodong Lin, and Yong Shi.
Active learning from stream data using optimal weight classifier ensemble.
IEEE Transactions on Systems, Man, and Cybernetics, Part B: Cybernetics, 40(6):1607 – 1621, 2010.

[38] Indrė Zliobaitė.
Learning under concept drift: an overview.
Technical report, Vilnius University, 2009.

[39] Indrė Zliobaitė, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes.
Active learning with evolving streaming data.
In *Proceedings of the 21st European Conference on Machine Learning and Principles and Practice of Knowledge Discovery in Databases (ECML PKDD'11)*, volume 6913 of *Lecture Notes in Computer Science*, pages 597–612. Springer, 2011.

[40] Indre Zliobaite, Albert Bifet, Bernhard Pfahringer, and Geoffrey Holmes.
Active Learning With Drifting Streaming Data.
IEEE transactions on neural networks and learning systems, 25(1):27–39, 2014.

Bibliography X

[41] Indrė Zliobaitė, Mykola Pechenizkiy, and João Gama.
An overview of concept drift applications.
In *Big Data Analysis: New Algorithms for a New Society*, volume 16 of *Studies in Big Data*, pages 91–114. Springer, 2016.