On-line Active Learning from Data Streams

Dr. Edwin Lughofer
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University, Linz, Austria
http://www.flll.jku.at/staff/edwin
“In active learning, students participate in the process and students participate when they are doing something besides passively listening. It is a model of instruction or an education action that gives the responsibility of learning to learners themselves.”

Basic idea has been transferred to Machine Learning and Data Mining Community (Cohn, 1994) where the main idea was to give responsibility to the model itself with which data samples it should be further learned in order to be (self-)improved.
Overview

- Motivation and Problem Discussion
- Principal WorkFlow of AL Scenarios
- **State-of-the-Art 1**: Overview of Methods for Adaptive Models (based on Classical Incr. Learning)
- **State-of-the-Art 2**: On-line Active Learning with Evolving (Fuzzy) Systems
  - Basics of Evolving Modeling from Streams
  - Single pass selection concepts for Evolving Classification Problems
  - Single pass selection concepts for Evolving Regression Problems
- Applications of On-line Active Learning
Data Streams – Characterization
(Gama, 2010)

- The data samples or data blocks are continuously arriving online over time
  - Requires fast update of model parameters and structures

- The data samples are arriving in a specific order, over which the system has no control.
  - Requires robust evolving methods

- Data streams are usually not bounded in a size
  - Requires open-loop adaptivity, incremental learning

- Once a data sample/block is processed, it is usually discarded immediately, afterwards
  - Requires single-pass learning capability

- Changing characteristics over time (target concept change, several types of drifts)
  - Requires increased flexibility (forgetting of older concepts)
Adaptive Models - Characterization

- **Incrementality:** accounts for a step-wise (sample or block) processing of data and model building over time
- **Single-Pass Capability:** omits time-intensive re-training cycles; new sample/buffer is loaded, sent into the incremental learning engine and discarded, afterwards.

- **Adaptivity:** accounts for (recursively) adapting parameters and structures with newly loaded samples within incremental learning steps
- **Flexibility:** forgetting, outdating of older learned concepts over time
Evolving ≠ Evolutionary

Genetic operators working in fully batch mode, iteratively over the same data set multiple times
Evolving Systems - Characterization

- Adaptive Models Characteristic (see previous slide)

  + **Evolving property**: structural components (e.g., rules, neurons, leaves) are added on demand and on the fly due to new system states, operating conditions etc.

  => Real knowledge expansion takes place
“Adaptive vs. Evolving” Illustration

Adaptive Case
- Old cluster contour
- Updated cluster contour
- Case 1: new samples fit into the structure => model (cluster) refinement
- Old cluster contour
- New cluster contour
- ... Old Samples
- ... New Sample(s)

Evolving Case
- Old cluster contour
- New cluster contour
- Case 2: difference in data distribution => evolve a new structural component (cluster)
- Case 3: new data denoting new knowledge => evolve a new structural component (cluster)
Motivation for On-line AL (even stronger than in batch case)

- **Similar reasons** as in case of classical, batch processing

- **+ Necessity for active learning** becomes even more demanding because:
  - Models **have to be updated regularly**, typically in fast manner with low lags, in order to cope with changing system dynamic and non-stationary environments (reflected in the streams) and with the real-time demands => target values required to avoid unsupervised adaptation
  - **Target values may be cost-intensive** to measure (sensor wear) or to retrieve from users/experts (classification case)
  - **Getting feedback from operators** is even more expensive and uneconomic than in the batch, off-line case: user/operator is disturbed from his main business significantly when feedback on each single sample is requested!
Principal Workflow for On-line (vs. Off-line) AL

Batch Off-line Active Learning Workflow

- Pool of Unlabelled Data Samples
- Off-line Active Sample Selector
- Selected Data
- Retrain Model
- New Samples to be Selected??
  - Yes
  - No
  - End

On-line Active Learning Workflow

- New Data from Stream
  - Predict Target Value
    - On-line Active Selection
      - Sample Selected??
        - Yes
        - AL Buffer full?
          - Yes
          - Operator / Expert
            - True Sample Labels
          - No
          - Update Model, Empty Buffer
        - No
          - No
          - No
# Methods for Adaptive Models - Overview

State-of-the-art methods for online active learning and their properties (as indicated in the columns).

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>PER-AL-Ext [16]</td>
<td>Single-pass</td>
<td>C (bin)</td>
<td>Probability (fixed or adaptive)</td>
<td>Perceptron (linear)</td>
<td>No</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>LO-SVMs [83]</td>
<td>Re-training</td>
<td>C (bin)</td>
<td>Uncertainty (poorly classified)</td>
<td>SVMs (linear)</td>
<td>Yes</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td>LASVM [7]</td>
<td>Window (random samples)</td>
<td>C (bin)</td>
<td>Three variants (grad., distance, random search)</td>
<td>SVMs (linear)</td>
<td>No</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td>OAL-BP [18]</td>
<td>Single-pass</td>
<td>C (bin)</td>
<td>Probability (fixed)</td>
<td>Bayesian (linear)</td>
<td>No</td>
<td>Dynamic (based on func. value)</td>
<td>None</td>
</tr>
<tr>
<td>Slide-NN [41]</td>
<td>Window</td>
<td>C (multi)</td>
<td>Uncertainty (distance to boundary)</td>
<td>Nearest Neighbors</td>
<td>Yes</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td>OAL-IC [25]</td>
<td>Re-training</td>
<td>C (multi)</td>
<td>Committee-Based (different sol. strategies)</td>
<td>Any type</td>
<td>No</td>
<td>None</td>
<td>None</td>
</tr>
</tbody>
</table>

## Properties
- **Data Pre-Processing**: single-pass vs. sliding windows vs. re-training
- **Regression or Classification Approach**? => all are for Classification, 2 for multi
- **Sampling Type**: uncertainty vs. probability vs. committee-based
- **Model Architecture**: perceptron, SVMs, Baysian, nearest neighbor
- **Decremental AL embedded**?
- **Thresholding**: fixed vs. dynamic vs none
- **Distribution assumption**: 1 requires uniform samples, but all others have no assumption
**Incrementally Adaptive Methods (Examples)**

- **PER-AL** = *Perceptron-Based Active Learning*
  - Linear classifier
  - AL based on uncertainty, measured in terms of \( \text{abs}(v_t x) \)
  - Update rule according to the "Reflection" concept

- **Per-AL-Ext** = *Extended Version of above*
  - Four different variants for classifier update
  - Bernoulli-based sample selection
  - \( b/(b + |p_t|), \quad p_t = v_t \cdot \vec{x} \)

- **LO-SVMs** = *Linear On-line SVMs*
  - Re-training of SVMs which may be slow
  - Three selection variant: logistic + fixed margin sampling

- **LASVM** = *Linear Adaptive Support Vector Machines*
  - On-line kernel classifier with adaptive support vectors (but no evolution of new ones, selection by strongest minimax gradient)
Incrementally Adaptive Methods (Examples)

OAL-DSC = On-line Active Learning with Dynamic Selection Criteria
Z. Ferdowsi, R. Ghani, and R. Settimi, ICDM, 2013

New Data Chunk

- ISS 1 (uncertainty)
  - Selected Samples ISS 1
  - Train Classifier 1

- ISS 2 (density)
  - Selected Samples ISS 2
  - Train Classifier 2
  - Calculate Mean Score on Top k% from „New Data Chunk“
  - Elicit Classifier with highest Mean Score Overall (1-4)

- ISS 3 (hybrid)
  - Selected Samples ISS 3
  - Train Classifier 3

- ISS 4 (sparsity)
  - Selected Samples ISS 4
  - Train Classifier 4
  - Show Selected Samples by corresponding ISS to Operator for Labeling

Labelled Samples seen so far

Current Classifier

Can be Any Type of Classifier

Dr. Edwin Lughofe  http://www.flll.jku.at/staff/edwin
Methods for Evolving (Fuzzy) Models

Dr. Edwin Lughofer
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University, Linz, Austria
http://www.flll.jku.at/staff/edwin
Current Adaptive Methods address uncertainty in two ways:

- Closeness to the decision boundary (based on expected error region, likelihood for a certain output class, ...)
- Disagreement degree among model ensembles

Uncertainty due to Novelty Content is hardly addressed

=> AL with Evolving Models to address Novelty Content in a Natural Way (due to their explorative nature)

=> Recall Figure from before
Evolving Modeling Key Steps

Key Steps in an Evolving Modeling Engine

(1) Load new data sample $\tilde{x}$
(2) Pre-process data sample (e.g. normalization)
(3) If model structure is empty, initialize the first component with its center to the data sample $\tilde{c} = \tilde{x}$ and its spread (range of influence) to some small value; goto Step (1).
(4) Else, perform the following steps (5-10):
(5) Check if component evolution criteria are fulfilled (this corresponds to knowledge expansion and makes it evolving):
   (a) If yes, evolve a new component and perform body of Step (3) (without if-condition).
   (b) If no, proceed with next step.
(6) Set forgetting factor appropriately to account for the current system dynamics (reflected in drifts in the data); this steers the flexibility of subsequent model updates (high forgetting $\rightarrow$ high model flexibility, no forgetting $\rightarrow$ classical lifelong learning).
(7) (only in case of fuzzy models: update antecedents parts of (some or all) components).
(8) Update parameters of (some or all) components.
(9) Check if the component pruning/merging criteria are fulfilled (this corresponds to knowledge contraction and makes it evolving, too):
   (a) If yes, prune or merge components; goto Step (1).
   (b) If no, proceed with next step.
(10) Optional: Perform corrections of parameters towards more optimality.
(11) Goto Step (1).
Model Architecture Employed

- **Fuzzy rule-based models:**
  - Universal Approximators
  - Strong Support of Many EFS Approaches in Literature
  - Offering Linguistic Insights into System Behaviors / Models

Example for Residential Premise Prices

**Figure 1.** Fuzzy partitions for the five input variables (a) to (e): Area, Age, Storeys, Rooms, Centre and the output variable Price ($).

**Figure 2.** Fuzzy partitions for the two input variables Storeys and Rooms when not using any merging/pruning option in FLEXFIS – compare with those in Figure 1.

**Readable rules**

1. Rule 1: If Area is LOW and Age is NEW and Storeys is AV and Rooms is FEW and Centre is CLOSE Then Price is LOW
2. Rule 2: If Area is LOW and Age is VeryNEW and Storeys is MANY and Rooms is FEW and Centre is CLOSE Then Price is LOW
3. Rule 3: If Area is LOW and Age is NEW and Storeys is AV and Rooms is USUAL and Centre is CLOSE Then Price is LOW
4. Rule 4: If Area is LOW and Age is OLD and Storeys is AV and Rooms is FEW and Centre is CLOSE Then Price is LOW
5. Rule 5: If Area is LOW and Age is NEW and Storeys is FEW and Rooms is FEW and Centre is FAR Then Price is LOW
6. Rule 6: If Area is LOW and Age is MEDIUM and Storeys is MANY and Rooms is FEW and Centre is VeryCLOSE Then Price is MEDIUM
7. Rule 7: If Area is MEDIUM and Age is NEW and Storeys is FEW and Rooms is USUAL and Centre is MEDIUMThen Price is MEDIUM
8. Rule 8: If Area is HIGH and Age is MEDIUM and Storeys is AV and Rooms is MANY and Centre is CLOSE Then Price is HIGH
9. Rule 9: If Area is MEDIUM and Age is MEDIUM and Storeys is AV and Rooms is USUAL and Centre is CLOSE Then Price is HIGH
Fuzzy Rules Definition for Regression

Rule \(_i\) : IF \((x_1 \text{ IS } \mu_{i1})\) AND...AND \((x_p \text{ IS } \mu_{ip})\) THEN \(l_i \text{ IS } \Phi_i\)

- **Mamdani** (Mamdani, Assilian, FSS, 1977)
  - \(\Phi_i\) : fuzzy set

- **Sugeno**
  - \(\Phi_i\) : singleton numerical (real) value

- **Takagi-Sugeno** (Takagi and Sugeno, IEEE SMC, 1985)
  - \(\Phi_i\) : linear function (hyperplane)

- **Takagi-Sugeno-Kang** (Sugeno and Kang, FSS, 1988)
  - \(\Phi_i\) : polynomial functions, Gamma, Kernels (local SVM)

- **Generalized Takagi-Sugeno** – non-axis parallel contours! New development in Lemos Gomide et al., IEEE TFS, 2011
  - Antecedent part is a multidimensional kernel, e.g. \(\exp(-{(X - C_i)^T\Sigma_i^{-1}(X - C_i)})\)
Fuzzy Rules Definition for Classification

- **Definition of Classical Fuzzy Rules:**

  \[
  \text{IF } x_1 \text{ is HIGH and } x_2 \text{ is INTENSE and } x_3 \text{ is LOW,}
  \]
  \[
  \text{THEN } \text{Class} = 3,
  \]
  
  **Linguistic Terms:** formally represented by fuzzy sets => granulation into local regions (one rule defines a local region in the feature space where Class = 3 is most likely)

- **Definition of Extended Fuzzy Rules:**

  \[
  \text{IF } x_1 \text{ is HIGH and } x_2 \text{ is INTENSE and } x_3 \text{ is LOW,}
  \]
  \[
  \text{THEN } \text{Class} = 1 \ (conf_1), \text{Class} = 2 \ (conf_2), \ldots, \text{Class} = K \ (conf_K),
  \]
  
  **Certainty information:** can indicate class overlaps in rule (local region) when being similar among classes

- **Also Used in All-Pairs Scheme** (bin. class-pairs)
All-Pairs (AP) EFC for Multi-Class Classification Problems (Lughofer and Buchta, IEEE TFS, 2013)

- Idea: Decomposition of (large-scale) multi-classification problem into several \((K^*(K-1)/2)\) smaller binary classification problems

\[
C_{k,l} \leftarrow T_{k,l}(X_{k,l}) \quad X_{k,l} = \{x|L(x) = k \lor L(x) = l\}
\]

Training procedure

- \(=>\) Decision Boundaries easier and faster to learn (theoretical complexity for updates lower) (Lughofer and Buchta, IEEE TFS, 2013)

- \(=>\) AP EFC may out-perform Single Model EFC and One-vs-Rest Multi-Model EFC (Angelov and Lughofer, 2008) by up-to 10% classification accuracy (Lughofer and Buchta, IEEE TFS 2013)

- Preference relation matrix: storing preference degrees for class \(k\) over \(l\), \(k,l=1,...,K, k\neq l\) for query point

- Final class response by scoring:

\[
L = \arg\max_{k=1,...,K} \left( \text{score}_k = \sum_{i^* = 1,...,[K/n]} \text{conf}_{k,i^*} \right)
\]

Reciprocal confidences:

\[
\text{conf}_{k,l} = 1 - \text{conf}_{l,k}
\]
Based on **Conflict and Ignorance Concepts**:

**Version Space**

- **Class Overlapping Case within Rule**, $conf_1 = 0$, $conf_2 = 4/9$, $conf_3 = 5/9$

- **Query #2, Conflict** (close to dec. Boundary)
- **Query #3, Conflict** (sign. overlap within a rule)
- **Query #4, Ignorance**

- **Clean Rule**

**Decision Boundary for Direct Multi-Class (highly non-linear)**
- **Decision Boundaries for Binary Class Pairs Decompos. (less complex)**

**For All-Pairs:**

$$\text{conflict}_{deg} = \frac{\text{score}_k}{\text{score}_k + \text{score}_1}$$

- **Membership degree to nearest rule with majority class $m$**

- **Membership degree to second nearest rule with different majority class $m^*$**

$$\text{conflict}_{deg} = 1 - \max_{k=m,m^*} (conf_k) = \frac{\mu_1(x)h_{1,k} + \mu_2(x)h_{2,k}}{\mu_1(x) + \mu_2(x)}$$

- **$h_{1,k} = \frac{h_{1,k}}{h_{1,m} + h_{1,m^*}}$**
- **$h_{2,k} = \frac{h_{2,k}}{h_{2,m} + h_{2,m^*}}$**

**ign. degree**

$$\text{ign. degree} = 1 - \max^c_{i=1} \mu_i(x)$$
**Version Space Reduction Example**

**Definition 2** Ignorance belongs to that part of classifiers uncertainty that is due to a query point falling within the extrapolation region of the feature space.

Updating the classifier with such samples extends the classifiers range of influence and decreases the variability of the version space.
Extended Confidence Calculation Expressing Conflict

- Problem of conv. Confidence calculation: not resolving any conflict between two rules
- \( \Rightarrow \) Extended confidence calculation (\( L = \) output class)

\[
L = k^* \quad \text{with} \quad k^* = \arg\max_{1 \leq k \leq K} \left( \frac{\mu_1(x)h_{1,k} + \mu_2(x)h_{2,k}}{\mu_1(x) + \mu_2(x)} \right)
\]

\[
\text{conf}_L = \frac{\mu_1(x)h_{1,L} + \mu_2(x)h_{2,L}}{\mu_1(x) + \mu_2(x)}
\]

\[
h_{1,L} = \frac{h_{1,L}}{h_{1,L} + h_{1,L_2}} \quad h_{2,L} = \frac{h_{2,L}}{h_{2,L} + h_{2,L_2}}
\]

Conf\(_L\) lies in \([0.5,1]\)

Maximal conflict (binary problems)

Maximal certainty\( \Rightarrow \) Update on all samples for which conf\(_L\) < thresh

Nearest rule \( \Rightarrow \) confidence = 1

Second nearest rule with majority class label

\( L_2 \neq L \)

Maximal conflict

\( L = \) Rect.

\[
\begin{align*}
\mu_1 & \sim \mu_2 \\
h^*_{1,L} &= 1 \\
h^*_{2,L} &= 0 \\
\Rightarrow \quad \text{Conf}_L &\sim 0.5 \quad \text{(max. conflict)}
\end{align*}
\]

\( L = \) Rect.

\[
\begin{align*}
\mu_1 & > \mu_2 \\
h^*_{1,L} &= 1 \\
h^*_{2,L} &= 0 \\
\Rightarrow \quad \text{Conf}_L &>> 0.5 \quad \text{(no conflict)}
\end{align*}
\]
**Behaviour Decision Boundary**

Example: two nearest rules **having different majority classes** =>
decision boundary inbetween, query point with high conflict

Example: two nearest rules **having same majority class**
=> decision boundary between these two and a third rule with
different majority class
=> query point with low conflict
Based on three Single-Pass Criteria:

- **Ignorance in the Input Space** using as a combined criterion of extrapolation and non-linearity degree
- **Uncertainty in predictive model outputs** (high confidence intervals)
- **Uncertainty in model parameters**

Ignorance Concept: Novelty Content $\Leftrightarrow$ *Extrapolation Degree* $+$ *Non-linearity Degree*
Uncertainty Sampling based on Novelty Content (Feature Space Exploration)

- **Specific Criterion for Extrapolation** in case of multi-variate Gaussians – motivated by Statistical Theory:
  \[
  \min_{i=1,\ldots,C} \sqrt{((\bar{x} - \bar{c}_i)^T \Sigma^{-1} (\bar{x} - \bar{c}_i))} > r_i
  \]

- **Combined Criterion**
  \[
  r_i = \text{fac} \times \left(p^{1/\sqrt{2}}\right)
  \]

- **Non-linearity Degree Approx. Measure**
  \[
  \text{deg}_{\text{nonlin}} = \sum_{i=1}^{C} (1 - \frac{\phi(i, k)}{\pi}) + 1
  \]
  \[
  \phi(i, k) = \arccos \left( \frac{\bar{a}^T \bar{b}}{||\bar{a}|| \cdot ||\bar{b}||} \right)
  \]

  \[k = \arg \min_{j=1,\ldots,C/i}(d(\bar{c}_i, \bar{c}_j))\]

  Angle between adjacent hyperplanes – measures the actual non-linearity degree (# of rules may be very inexact!)

  *fac = variation parameter, steering # of selected samples
Example of Uncertainty in Model Outputs

- **Low local data density** (rule having 1 sample)
  - Prediction very certain
  - High uncertainty in prediction, but using a sample falling into this for model update will not further contract the wideness of the error bars!
  - Support of the local rule is low!

- **High (local) data density**
  - Prediction very uncertain
  - High uncertainty in prediction, but using a sample there for model update will increase model significance
  - But support of the local rule is high!

Prediction everywhere certain => no need for sampling
Uncertainty Sampling based on Prediction Confidence (Model Certainty Intensification)

- Modeled by **local error bars** (leaned on Skrjanc, CILS, 2009)
- Serving as **statistical confidence intervals** surrounding the model with different widths in **different local regions** (deduced from statistical noise and quantile estimation theory):

\[
\hat{f}_j(\bar{x}_k^*) = \Psi_j(\bar{x}_k^*) l_j(\bar{x}_k^*) \pm t_{\alpha, \Sigma(N) - deg} \tilde{\sigma} \sqrt{ (\bar{x}_k^* \Psi_j(\bar{x}_k^*))^T P_j(\Psi_j(\bar{x}_k^*) \bar{x}_k^*) } 
\]

- Extending to local variance of model errors:

\[
\hat{\sigma}_j(N) = \frac{1}{\Sigma - deg} \sum_{k=1}^{N} \Psi_j(\bar{x}_k^*)(e_k - \bar{e}_j)^2 \\
\Sigma(N) = \sum_{k=1}^{N} \Psi_j(\bar{x}_k^*)
\]
Uncertainty Sampling based on Parameter Uncertainty (Model Stability Intensification)

- Based on a **decrease on the A-optimality criterion** (trace of the inverse of the Fisher information matrix)

\[
\max_{i=1,\ldots,C} \left( \frac{\text{trace}(F_i^{-1})(k) - \text{trace}(F_i^{-1})(k+1)}{\text{trace}(F_i^{-1})(k)} \right) > \text{thresh}
\]

For linear consequent parameters it is the **inverse Hessian matrix** $P_i$ as updated through recursive fuzzily weighted least squares

\[
\begin{align*}
\omega_i(k+1) &= \tilde{\omega}_i(k) + \gamma(k)(y(k+1) - \tilde{r}^T(k+1)\tilde{w}_i(k)), \\
\gamma(k) &= P_i(k+1)\tilde{r}(k+1) = \frac{P_i(k)\tilde{r}(k+1)}{\frac{\lambda}{\Psi_i(x(k+1))} + \tilde{r}^T(k+1)P_i(k)\tilde{r}(k+1)}, \\
\end{align*}
\]

\[
P_i(k+1) = \frac{1}{\lambda} (I - \gamma(k)\tilde{r}^T(k+1))P_i(k),
\]

Per rule separately

Fuzzy sample weights

Forgetting factor
Sliding-Window Based Incremental and Decremental AL

(Cernuda, Lugofer et al., Chemometrics and Intelligent Laboratory Systems, 2014)

- Initial Window
- Update Chemo Model
- Retrain Model

Update Model (Unsupervised, permanent)

Create Model

Predict N next samples

Update Window (Supervised, max. 1 time per batch)

Ingoing sample selection:
- Static (fixed selection ratio)
- Dynamic (Active Learning)

Outgoing sample selection:
- Oldest out (naive approach)
- Less-info out (Active Learning)

Sliding Window based retrain

Ingoing sample selection:

Outgoing sample selection:

Samples in the current window

Outgoing sample

If none is similar according to the Bayesian similarity measure

Leave-one-sample-out (LOSO)

Regions of coverage

For spectra data

\\[ Q_\alpha = \theta_1 \cdot \left( \frac{c_a \sqrt{2 \theta_2 h_0^2}}{\theta_1} + 1 + \frac{\theta_2 h_0 (h_0 - 1)}{\theta_1^2} \right)^{1/h_0} \]

\\[ T_{a,m,\alpha} = \frac{a(m-1)}{m-a} \cdot F_{a,m-a,\alpha} \]
Alternative Approaches

- **Sliding window-based with random forests (Slide-AL-RF)**
  (Weigl, Lughofer 2016 et al.)
  \[ c = P_{\text{max}1} - P_{\text{max}2} \]
  \[ p_i = \frac{|\{L_b = i | b = 1, ... m\}|}{m} \]
  Low means that probabilities of two most likely classes are equal

- **Rclass:** What-to-learn concepts based on an **evolving, type-2 fuzzy classifier** (using the spherical potential) (Pratama, Anavatti 2015)
  - Two stage adaptation: pure parameter and parameter + rule evolution
  - Restriction by a budget constraint

- **SB-AL Disagreement degree of two hypothesis**, drawn from the same sample distribution, in their class. statements
  - Evolution of Gaussians when both hypothesis put samples into unknown class

  (1) Receive an instance \( x_n \in \mathbb{R}^d \) from the data stream.
  (2) Draw two random hypotheses \( h_1 \) and \( h_2 \) from the model posterior to form a committee.
  (3) For each hypothesis, compute the posterior \( p(c_n | x_n) \) under the Pitmann-Yor versus Diriclet Process (PYP) assumption [70].
  (4) Query the instance if the two hypotheses disagree on its classification, or they both assign the instance to an unknown class.
  (5) Include the labelled \( \{(x_n; c_n)\} \) in the training set \( L \) to refine the classifier
  (6) The iteration stops when a criterion is met, e.g. the query budget is exhausted.
# Overview over all Methods

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>Adaptive Approaches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(exp. error region)</td>
<td>(linear)</td>
<td></td>
<td>(error region shrink)</td>
<td></td>
</tr>
<tr>
<td>CBGZ [16]</td>
<td>Single-pass</td>
<td>C (bin)</td>
<td>Probability</td>
<td>Perceptron</td>
<td>No</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(fixed or adaptive)</td>
<td>(linear)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LO-SVMs [83]</td>
<td>Re-training</td>
<td>C (bin)</td>
<td>Uncertainty</td>
<td>SVMs</td>
<td>Yes</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(poorly classified)</td>
<td>(linear)</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASVM [7]</td>
<td>Window (random samples)</td>
<td>C (bin)</td>
<td>three variants</td>
<td>SVMs</td>
<td>No</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(grad., distance, random search)</td>
<td></td>
<td></td>
<td>(best out of window)</td>
<td></td>
</tr>
<tr>
<td>OAL-BP [18]</td>
<td>Single-pass</td>
<td>C (bin)</td>
<td>Probability</td>
<td>Bayesian</td>
<td>No</td>
<td>Dynamic</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(fixed)</td>
<td>(linear)</td>
<td></td>
<td>(based on func. value)</td>
<td></td>
</tr>
<tr>
<td>Slide-NN [41]</td>
<td>Window</td>
<td>C (multi)</td>
<td>Uncertainty</td>
<td>Nearest Neighbors</td>
<td>Yes</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(distance to boundary)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>OAL-IC [25]</td>
<td>Re-training</td>
<td>C (multi)</td>
<td>Committee-Based</td>
<td>Any type</td>
<td>No</td>
<td>None</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(different sol. strategies)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Evolving Approaches</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP-AL-C [48]</td>
<td>Single-Pass</td>
<td>C (multi)</td>
<td>Uncertainty + Novelty</td>
<td>EFC-SM/EFC-AP</td>
<td>No</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Conflict + Ignorance)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>SP-AL-R [56]</td>
<td>Single-Pass</td>
<td>R</td>
<td>Uncertainty + Novelty</td>
<td>EFS (gen. TS)</td>
<td>No</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Structural + output uncert.)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Slide-AL-RF [96]</td>
<td>Re-training</td>
<td>C (Multi)</td>
<td>Committee-based</td>
<td>Random Forests</td>
<td>No</td>
<td>Fixed</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(confidence, disagreement)</td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(statistical, Q and T^2)</td>
<td></td>
<td></td>
<td>(Statistical)</td>
<td></td>
</tr>
<tr>
<td>McIT2FIS [91]</td>
<td>Buffer (of past selected)</td>
<td>C (Multi)</td>
<td>Error + Novelty</td>
<td>Type-2 Neuro-Fuzzy Classifier</td>
<td>No</td>
<td>Fixed + Adapt.</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(output + distance)</td>
<td></td>
<td></td>
<td>(Several ones)</td>
<td></td>
</tr>
<tr>
<td>rClass [71]</td>
<td>Single-Pass</td>
<td>C (Multi)</td>
<td>Uncertainty</td>
<td>recurrent TSK fuzzy classifier</td>
<td>No</td>
<td>Adaptive</td>
<td>None</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(Conflict)</td>
<td></td>
<td></td>
<td>(Budget constraint)</td>
<td></td>
</tr>
<tr>
<td>SB-AL [46]</td>
<td>Single-Pass</td>
<td>C (Multi)</td>
<td>Committee-Based + Novelty</td>
<td>Gaussian mixture models</td>
<td>No</td>
<td>None</td>
<td>multivar. normal</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>(New classes add)</td>
<td></td>
<td></td>
<td>(Budget constraint)</td>
<td></td>
</tr>
</tbody>
</table>

Dr. Edwin Lughof | http://www.fll.jku.at/staff/edwin
Three Real-World Applications
(on-line AL concepts fully implemented and evaluated)

Dr. Edwin Lughofer
Department of Knowledge-Based Mathematical Systems
Johannes Kepler University, Linz, Austria
http://www.flll.jku.at/staff/edwin
On-line Visual Inspection of Micro-Fluid Chips
(Weigl, Lugofer et al., MVA, 2015)

Red = new components dealing with AL and Classifier Updates

Event Types during Visual Inspection, Recognized/Segmented

Production

Feature Vector Extraction

Evolving (Fuzzy) Classifier

Update (Fuzzy) Classifier, Integrate New Classes On-The-Fly

Sample Selection with Active Learning

Active Learning Buffer

Input to

Overwrite Classifier

Feat. Vector + Prediction

Drift Detection

Classification / Decision

Production Line (In-line Process)

Labels of Samples, New Classes

Warnings to Operator

If AL Buffer not filled

Explain Classifier Decisions (on Samples in AL buffer)

If AL Buffer Filled
Some Results
(Accumulated Accuracies with Reduced Samples)

Micro Fluid Chips with Active Learning

Comparison of EFC-AP (left) with Random Forests
Re-trained (right)

-Laufrollen with Active Learning

Comparison EFC-SM (left) with EFC-AP (right)
Examinining the Question whether actively selected samples are sufficient for drift detection in *data stream classification problems* …. (Lughofer, Weigl et al. 2016)

<table>
<thead>
<tr>
<th>Intens. / %</th>
<th>Supervised</th>
<th>Superv. Slided</th>
<th>Semi-Superv. Slided</th>
<th>Unsuperv. Slided</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Yes/No</td>
<td>Delay</td>
<td>False</td>
<td>Yes/No</td>
</tr>
<tr>
<td>15 / 50%</td>
<td>yes</td>
<td>510</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>15 / 33%</td>
<td>yes</td>
<td>232</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>15 / 20%</td>
<td>yes</td>
<td>266</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>15 / 10%</td>
<td>yes</td>
<td>250</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>30 / 50%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>30 / 33%</td>
<td>yes</td>
<td>253</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>30 / 20%</td>
<td>yes</td>
<td>269</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>30 / 10%</td>
<td>yes</td>
<td>411</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>45 / 50%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>no</td>
</tr>
<tr>
<td>45 / 33%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>45 / 20%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>45 / 10%</td>
<td>yes</td>
<td>275</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>60 / 50%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>no</td>
</tr>
<tr>
<td>60 / 33%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>60 / 20%</td>
<td>no</td>
<td>NA</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>60 / 10%</td>
<td>yes</td>
<td>281</td>
<td>0</td>
<td>yes</td>
</tr>
<tr>
<td>Sum/Av.</td>
<td>yes (9)</td>
<td>305</td>
<td>0</td>
<td>yes (16)</td>
</tr>
</tbody>
</table>

All samples used for drift detection (modified PH-test)  
Actively Selected Samples used => similar performance in delay
Active Learning Results in On-line Melamin Resin Production (Sliding Window Based Approach)

No Active Sample Selection, blind equidistant adaptation

Active Sample Selection

Savings: ~87% of measurement Costs with Titration Automat! (~13% Samples used for Selection)
Active Learning Results in On-line Viscose Production (Sliding Window Based Approach)

- Error Distributions over the complete Streams for different decremental active learning methods: random deletion from the window, oldest out and selected by AL („LessInfo“)

More high errors without decremental AL
Thanks a lot for Your Attention !!