Learning in Nonstationary Environments

Intelligence for Embedded Systems
Ph. D. and Master Course
Manuel Roveri
Politecnico di Milano, DEIB, Italy
Up to now we assumed the system model to be time invariant…
But everything and everybody changes over time ...

Be aware of *Gradual Concept drift*...
Everything and everybody changes over time ...

... and of *Aprupt* Concept drift
Learning in Nonstationary Environments: the effect of the non-stationarity

- Faults
- Ageing effects
- Changes in the environment

Estimate a model

Data generating process

P

(x,y)

Application

Obsolete model

\[
p(x|t) = \sum_{y \in \Lambda} p(y|t)p(x|y, t),
\]

\[
y(k) = \sum_{t=0}^{T-1} y(t)
\]

\[
A(z)y(k) = \sum_{i=1}^{m} \frac{C(z)}{D(z)} d(k)
\]

Perturbed, incorrect and missing data can hence heavily affect the subsequent processing phase so as to possibly induce wrong decisions or on-the-field reactions.
Stationarity and time invariance

- **Stationarity**
  - We say that a data generating process is stationary when generated data are i.i.d. realizations of a unique random variable whose distribution does not change with time.

- **Time invariance**
  - We say that a process is time invariant when its outputs do not explicitly depend on time.

\[ y(t) = a_1(e^{t_0-t})y(t-1) + a_2y(t-2) + \eta, \eta = N(0, \sigma^2) \]
Searching for adaptation

- Traditional assumption: stationarity hypothesis
- Adaptive solutions in a non-stationary framework:
  - A comprehensive methodology addressing this problem is not available
**WHAT: Instance Selection**

- **The idea**: identifying the samples of the training set that are relevant to the current state of the process.

  ![Diagram](image)

  - The adaptive systems generally rely on a window over the most recent training samples to process the upcoming data.
  
  - **fixed window approach**: the length of the window is fixed a-priori by the user
  
  - **heuristic approaches**: adapt the window length over the latest samples to maximize the accuracy
**WHAT: Instance Weighting**

- **The idea**: *training samples are not removed from the training set of the system but all the training samples (suitably weighted) are considered.*

  - The training samples might be weighted according to
    - the **age**
    - the **relevancy to the current state** of the process in term of accuracy of the last batch of supervised data
**WHAT: Multiple Models**

- **The idea:** *the outputs of an ensemble of models are combined by means of voting or weighted mechanisms to form the final output*

All these systems include techniques for dynamically including new models in the system or deleting obsolete ones (i.e., pruning techniques aiming at removing the oldest model or the one with the lowest accuracy).
Critical analysis of the considered approaches

- **Instance selection**
  - \(\uparrow\): low computational-complexity
  - \(\uparrow\): reduced training set
  - \(\downarrow\): fixed windows or heuristics to adapt the window size
  - \(\downarrow\): forgetting mechanisms

- **Instance weighting**
  - \(\uparrow\): low computational-complexity
  - \(\uparrow\): availability of all the training samples for recurrent models
  - \(\downarrow\): heuristics to define the sample weights
  - \(\downarrow\): full training set

- **Multiple models**
  - \(\uparrow\): availability of a model for “each bunch of data”
  - \(\downarrow\): high computational-complexity
WHEN: active vs passive approach

- **Active solutions** rely on triggering mechanisms to identify changes in the process and react by updating the model
  - The most popular triggering mechanism is the change detection

- **Passive solutions** continuously adapt the model without the need to detect the change
  - Ensembles of models with the adaptation phase consisting in a continuous update of the weights of the fusion/aggregation rule and creation/removal of models
Passive learning

Online (incremental) learning

\[ V_N(\theta, \{(x_i, y_i)\}) = L(y_i, f(\theta, x_i)) \]

\[ \theta_{i+1} = \theta_i - \eta \frac{\partial L(y_i, f(\theta, x_i))}{\partial \theta} |_{\theta_i} \]

Batch learning

\[ Z_{n,i} = \{(x_i, y_i), (x_{i-1}, y_{i-1}), \ldots, (x_{i-n+1}, y_{i-n+1})\} \]

\[ \theta_{i+1} = \theta_i - \eta \frac{\partial V_N(\theta, Z_{n,i})}{\partial \theta} |_{\theta_i} \]

Ensemble learning

\[ y(x) = \sum_{i=1}^{k} w_i M_i(x) \]
The Oracle provides information about an event, e.g., the occurrence of concept drift.
WHEN: Triggering mechanisms

Change detection on the pdf of the inputs:

- ↑: monitoring the distribution of unlabeled observation
- ↓: this solution does not allow us for detecting changes that do not affect the distribution of observations

Change detection on the classification error

- ↑: reacting to changes when these directly influence its accuracy
- ↓: the need of supervised samples
The active learning framework within an evolving environment

- Reference Concept
- Concept Library
- Learning Phase
- Concept Drift Detection
- Operational Phase
- Feature Extraction
- Application
- Adaptation
- Detection, Information about concept drift
- Time occurrence

Feature for drift detection

<table>
<thead>
<tr>
<th>Samples</th>
<th>Temperature (°C)</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Sensor 1</td>
</tr>
<tr>
<td></td>
<td>Sensor 2</td>
</tr>
<tr>
<td></td>
<td>Sensor 3</td>
</tr>
</tbody>
</table>

Active learning framework within an evolving environment.
Features must be i.i.d (but data are generally signals)

- **Data space**
  - Raw data are used (e.g., the minimum of the water consumption of a day @ district metered area)
Features (2)

- Feature space

✓ Any i.i.d. feature (e.g., residual, measurements in a quality analysis applications)
Model space

- LTI models are used to approximate the signal

\[ y_1(t) \xrightarrow{\hat{f}_{\theta}^{1,2}} y_2(t) \]

Estimate model parameters

\[ \hat{\theta}_{t_1}, \ldots, \hat{\theta}_{t_n} \]
Active learning

Phase

Concept Drift Detection

Identification of the new state

Retrain the application

How

Change detection tests CDT CPM Change point methods

Determination of consistent data instances

Online, batch or full learning
Concept drift detection

Ad hoc triggers designed to detect changes by inspecting sequences of data or derived features

- **Data-based methods**
  - Limit checking
  - Binary threshold

- **Statistical-based methods**
  - Statistical Hypothesis tests
  - Change-Point Methods
  - Change detection tests
Limit checking

- Testing if a given (measured) variable exceeds (indicating a change) or not a known absolute limit.

$$y(t) \leq Y_{\text{lim}} \rightarrow F(t) = 0$$
$$y(t) > Y_{\text{lim}} \rightarrow F(t) = 1$$

- Variants:
  - Two limits, associated to different levels of safety.
  - Use of superior and inferior limits.
- Easy to implement.
- Too conservative (low change sensitivity).
Estimation of mean and variance

- The monitored variables are usually stochastic variables $Y_i(t)$ with a certain pdf in nominal condition:
  \[ \mu_i = E\{Y_i(t)\}; \quad \bar{\sigma}_i^2 = E\{[Y_i(t) - \mu_i]^2\} \]

Changes are then expressed by:

\[ \Delta Y_i = E\{Y_i(t) - \mu_i\} \quad \text{and} \quad \Delta \sigma^2 = E\{[\sigma_i(t) - \bar{\sigma}_i]^2\} \]

If the pdfs do not significantly overlap, one could use a fixed threshold based on $\sigma$, e.g., $\gamma = 2\sigma$

Ratio between the detection of small changes and false alarms
More powerful techniques need to be considered

**Statistical tests**

- **off-line**: fixed length sequence (after storing all data)
- **on-line**: at each time instant

- **Statistical hypothesis tests:**
  - Off-line
  - Control of FPs

- **Change detection tests**
  - On-line
  - No control of FPs
Statistical hypothesis tests

- Theory of statistics
- Testing one hypothesis ($H_0$) against one or more alternative hypotheses $H_1, \ldots, H_N$
  - $H_0$: null hypothesis (no change) $\Rightarrow Y$ in $Y_0$
  - $H_1, \ldots, H_N$: change hypothesis $\Rightarrow Y$ in $Y_1$
- **Decision**: Based on the assumption that the null hypothesis is true if no fault occurs, the null hypothesis is rejected and the alternate hypothesis is accepted if the sample of the random variable $Y$ falls outside the region of acceptance. Otherwise, $H_0$ is accepted and $H_1$ rejected
Regions of rejection and acceptance for a HT

The distribution of sample means if the null hypothesis is true (all the possible outcomes)

Sample means close to $H_0$: high-probability values if $H_0$ is true

Extreme, low-probability values if $H_0$ is true

$\mu$ from $H_0$

Extreme, low-probability values if $H_0$ is true
How to set the regions?
# Hypothesis tests: the literature

<table>
<thead>
<tr>
<th>Test family</th>
<th>Type (P/NP)</th>
<th>Change (Ab/Dr)</th>
<th>Entity under test</th>
<th>1D/ND</th>
<th>On-line/Off-line</th>
<th>Training Set /A priori information</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Z-test</strong></td>
<td>Statistical Hypothesis testing</td>
<td>Parameteric</td>
<td>Abrupt</td>
<td>Mean</td>
<td>1D</td>
<td>Off-line</td>
<td>Parameters</td>
</tr>
<tr>
<td><strong>t-test</strong></td>
<td>Statistical Hypothesis testing</td>
<td>Parameteric</td>
<td>Abrupt</td>
<td>Mean</td>
<td>1D</td>
<td>Off-line</td>
<td>None</td>
</tr>
<tr>
<td><strong>Mann-Whitney U test</strong></td>
<td>Statistical Hypothesis testing</td>
<td>Non Parameteric</td>
<td>Abrupt</td>
<td>Median</td>
<td>1D</td>
<td>Off-line</td>
<td>None</td>
</tr>
<tr>
<td><strong>Kolmogorov-Smirnov test</strong></td>
<td>Statistical Hypothesis testing</td>
<td>Non Parameteric</td>
<td>Abrupt</td>
<td>Pdf</td>
<td>1D</td>
<td>Off-line</td>
<td>None</td>
</tr>
<tr>
<td><strong>Kruskal-Wallis test</strong></td>
<td>Statistical Hypothesis testing</td>
<td>Non Parameteric</td>
<td>Abrupt</td>
<td>Median</td>
<td>1D</td>
<td>Off-line</td>
<td>None</td>
</tr>
</tbody>
</table>
Change point methods

CPMs inspect a sequence of data and check for concept drift

Given sequence

\[ \mathcal{X} = \{x(t), t = 1, \ldots, n\} \]

Produce a generic partitioning

\[ \mathcal{A}_\tau = \{x(t), \ t = 1, \ldots, \tau\}, \]
\[ \mathcal{B}_\tau = \{x(t), \ t = \tau + 1, \ldots, n\} \]

and

\[ \tau \text{ is a change point if } x(t) \sim \begin{cases} \mathcal{F}_0, & \text{for } t < \tau \\ \mathcal{F}_1, & \text{for } t \geq \tau \end{cases} \]

In practice

\[ \begin{cases} \text{The estimated change-point in } \mathcal{X} \text{ is } M & \text{ if } \mathcal{T}_M \geq h_{n,\alpha} \\ \text{No change-point identified in } \mathcal{X}, & \text{ if } \mathcal{T}_M < h_{n,\alpha} \end{cases} \]
Example

\[ x(t) \sim \begin{cases} 
    \mathcal{N}(0, 1), & \text{if } t < 350 \\
    \mathcal{N}(-1, 1), & \text{if } t \geq 350 
\end{cases} \]

With hypothesis test

\[ H_0 : \forall t, \ x(t) \sim \mathcal{F}_0 \]

\[ H_1 : \exists \tau \ x(t) \sim \begin{cases} 
    \mathcal{F}_0, & \text{if } t < \tau \\
    \mathcal{F}_1, & \text{if } t \geq \tau 
\end{cases} \]

For instance consider the Student t statistics for the means

\[ D_\tau = \sqrt{\frac{\tau(n-\tau)}{n}} \frac{\bar{A}_\tau - \bar{B}_\tau}{S_\tau} \]
Change point methods

Threshold e.g., $h_{500, 0.05} = 3.225$ provided by the CPM package.
Change-detection tests

- Change detection tests are methods designed to detect variations in the pdf of the process generating the data.

  - **Parametric approach**: knowledge of the pdf before and after the change
    - CUSUM test
    - Shiryaev-Robert test
  
  - **Nonparametric approach**:
    - CI-CUSUM test, NPCUSUM test
    - ICI-based change detection test
The CUSUM test

- $X = \{x_1, x_2, \ldots, x_N\}$ : $p_\theta(x)$

- The change at $t_0$ modeled as a transition from $\theta_0$ to $\theta_1$ (Hp: we keep the pdf structure)

- Measure a discrepancy at time time $t$: $s_t = \ln \frac{p_{\theta_1}(x_t)}{p_{\theta_0}(x_t)}$

- Evaluate the cumulative sum $S_t = \sum_{i=1}^{t} s_t$

- CUSUM identifies a change at time $t$ when $g_t = S_t - m_t \geq h|_{t}$
  with $m_t = \min_{1 \leq i \leq t}(S_i)$

Kulback-Leibler
The CI-CUSUM test

1. Observations $X = \{x(t), t = 1, \ldots, T\}, x(t) \in \mathbb{R}^d$
2. Partitions of $X$ into disjoint intervals $Y(s) = \{x(t), (\nu - 1) \cdot s \leq t < s \cdot \nu\}$
3. Extract the average feature vector $\varphi_y(s)$ (e.g., mean, var., kur., skew.) from each subsequence $Y(s)$
4. The pdf is gaussian from the central limit theorem
5. Estimate the null hypothesis $\Theta^0$ from $TS = \{\varphi_y(s), s \leq s_0\}$
6. Define $m$ alternative hypotheses $\{\Theta^j\}, j = 1, \ldots, m$ as “not being in $\Theta^0$“
7. Measure the discrepancy at time $s$ as
   $R_j(s) = \sum_{\tau=1}^{s} \ln \left( \frac{N_{\Theta^j}(\varphi_y(\tau))}{N_{\Theta^0}(\varphi_y(\tau))} \right)$, $j = 1, \ldots, m$
8. CI-CUSUM identifies a change at time $s$ if $g(s) = R_j(s) - \min_{1 \leq \tau \leq s} R_j(\tau) > h_j$
The ICI-based change detection test

- The test relies on a set of functions that transform the observations into Gaussian distributed features.
- ICI rule: a method for developing adaptive estimates for regression of functions from noisy observations (signal and image denoising).

The ICI rule, combined with a polynomial regression technique, assesses the stationary of the features (and hence of the process).
Particularly effective in detecting changes but ... 

How to increase promptness in detection still maintaining robustness w.r.t false positives?
The answer to the question “what happened?” is not enough ...

... Tell me: “when did it happen?”

"Apparently you collapsed when told the price of these ..."
Not only detection of the change, but also estimation of the time instant the process becomes nonstationary.

After the detection, we want an estimate $T_{\text{ref}}$ of $T^*$ by means of a refinement procedure.
Hierarchical CDT

A multivariate hypothesis test based on the Hotelling T-square statistics

\[ S = (\bar{F}^0 - \bar{F}^1)' \left( \frac{1}{n_0} + \frac{1}{n_1} \right) \Sigma^{-1} \left( \bar{F}^0 - \bar{F}^1 \right), \]
\[ \left( \frac{n_0 + n_1 - 2}{n_0 + n_1 - N - 1} \right) \mathcal{F}(N, n_0 + n_1 - N - 1), \]

Change-point methods: statistical tests able to assess whether a given data-sequence contains (or not) a change point

Second level change-detection test aiming at confirming (or not) the change hypothesis:

- Multivariate hypothesis test
- Change-point methods
Which data are consistent with the current status?

- Instances: between $T^*$ and $\hat{T}$

$T^*$ is unknown: use estimates $T_{ref}$ and $\hat{T}$
If concept drift is detected the whole framework is retrained

- Retrain the application
- Application
  - Detection trigger
  - Reference concept

\[ O_{T_0} \quad T_0 \quad T^* \quad T^{\text{ref}} \quad \hat{T} \]
An example: Just-in-Time Adaptive Classifiers
Just-in-Time Adaptive Classifiers

- Nominal Concept
- Hierarchical Concept Drift Detection
- Feature Extraction

- Recurrent Concepts
- Statistical Moments
- Sample Statistical moments, Classification error

- Adaptation

- JIT Classifiers
  - K-NN
  - SVMs
  - Neural networks

- Dynamic knowledge base management
  - Estimate of change time

- ICI-based CDT on the observations and the errors
  - Hypothesis tests, Change-Point Methods
Asymptotic optimality with JIT classifiers

JIT adaptive classifiers grant asymptotic optimality when the process generating the data is affected by a sequence of abrupt concept drift.

Dataset

Gaussian classes
Dealing with concept drift ...

\[ p(x|t) = p(\omega_1|t)p(x|\omega_1, t) + p(\omega_2|t)p(x|\omega_2, t) \]
The novel idea: extending the JIT classifier

**Two** CDTs are to assess if:
- The *pdf* of the *input* is *stationary*
- The *classification error* is *stationary*

**Adaptation phase** consists in:
- *Isolation of the current concept*
- Identification of *recurrent concepts*
- Training the classifier by exploiting all the *available supervised information*

**Flowchart**:
1. **Application**
2. **Supervised/Unsupervised data**
   - **CDT**
     - Stationary
     - Non-stationary
   - **Identify current concept**
     - **Recurrent**
       - Yes
       - Reactivate previous classifier
     - No
       - Define the new KB (Adaptation)
   - KB
Some final remarks …

✓ Being acquainted with learning techniques is a plus in everybody’s background.

✓ Most of time the we can assume that the process generating the data is time invariant. When it is not we need to pay attention…

✓ Learning in a changing environment must be considered and represents a key property intelligent systems should possess.
Let’s play with MATLAB

- Download the examples related to Lecture 6
- In the ZIP file:
  - Example 6_A.m
    - Adaptation of NN in nonstationary environments
  - Hierarchical ICI-based Change Detection Test
    - Detection of a change and estimation of the time instant the change started