CO3091 - Computational Intelligence and Software Engineering

Lecture 25

Image from: http://www.achievebreakthrough.com/wp-content/uploads/2015/08/falling-big-1.jpg

Principles of Continuous Learning

Leandro L. Minku

Overview

- Importance and motivation for continuous learning.
- What is continuous learning?
- Continuous learning algorithms.
- Adapting predictive models to changes.
- Evaluation of continuous learning algorithms.

Importance and Motivation

- Many real world applications suffer changes over time. We say that they operate in changing environments. E.g.:
    - credit card approval,
    - prediction of electricity prices,
    - software effort estimation,
    - etc.

Importance and Motivation

- Many real world applications suffer changes over time. We say that they operate in changing environments. E.g.:
  - credit card approval,
  - prediction of electricity prices,
  - software effort estimation,
  - etc.
- These changes may affect the accuracy of predictive models.
  - A model that was good in the past may present bad accuracy after a change.

What to do When There is a Change?

Continuous Learning

- Many applications produce additional data over time.
- If we can learn additional data over time, we can improve our models based on such data.
- We can also use strategies to adapt our predictive models to changes.
- Continuous learning: algorithms that can learn additional data that arrive over time.
    - Lifelong learning.
    - Incremental learning.

Supervised Learning

Training Data / Examples

x1 = age	x2 = salary	x3 = gender	…	y = good/ bad payer
18	1000	female	…	Good
30	900	male	…	Bad
20	5000	female	…	Good
…	…	…	…	…

→ Machine Learning Algorithm →

Predictive Model

New instance **x**
for which we want
to predict the output →

→ Prediction

Image from: http://www.theenergycollective.com/sites/theenergycollective.com/files/imagepicker/488516/2013%20predictions%20follow%20up.jpg
Image from: http://api.ning.com/files/jPrLLXuRuu5NQUBB0TWzUm-twYVH7ySIXAqNfxr1znmJjEDubI*jqwrO0EfEEgHXuDo41tbZK*Nk1rYqydv-PiWaKyjdpQt6/2013Predictions.jpg

Continuous Supervised Learning

Training Data Stream

x1 = age	x2 = salary	x3 = gender	…	y = good/ bad payer
…	…	…	…	…

→ Machine Learning Algorithm

↓

New instance **x** for which we want to predict the output →

Predictive Model

→ Prediction

Image from: http://www.theenergycollective.com/sites/theenergycollective.com/files/imagepicker/488516/2013%20predictions%20follow%20up.jpg
Image from: http://api.ning.com/files/jPrLLXuRuu5NQUBB0TWzUm-twYVH7ySIXAqNfxr1znmJjEDubI*jqwrO0EfEEgHXuDo41tbZK*Nk1rYqydv-PiWaKyjdpQt6/2013Predictions.jpg

Continuous Supervised Learning

Training and model usage co-occur.

The predictive model being used can be **updated** with new training examples.

Types of Continuous Learning Algorithms

- Online machine learning algorithms:
  - Update predictive models whenever a new training example is received.

- Chunk-based learning algorithms:
  - Update predictive models when a whole chunk of training examples is received.

We will concentrate on online algorithms, because they can also be used when training examples arrive in chunks.

Learning Machines That Can Be Used For Continuous Learning

- k-NN:
  - Original algorithm can be used for continuous (online) learning.
  - However, it may run into storage and efficiency problems as data become big.
  - Needs strategies to adapt to changes.

Learning Machines That Can Be Used For Continuous Learning

- Decision trees:
  - Original algorithms need whole training set to decide on splits.
    - If you don't have all the training data beforehand, you need to re-build the model from scratch to learn new data.
    - Problem of using past + new training data:
      - Inefficient and does not allow us to deal with changes.
    - Problem of using only new data:
      - It can waste useful knowledge.

Learning Machines That Can Be Used For Continuous Learning

- Decision trees:
  - Other versions have been created for continuous (online) learning. E.g.: Hoeffding trees.
    - Associate training examples to nodes as they arrive.
    - Nodes with less than a minimum number of training examples are made into leaf nodes.
    - When a leaf node reaches a given number of examples, split this node.
    - Most of the continuous learning versions of decision trees still need some strategy to adapt to changes.

Learning Machines That Can Be Used For Continuous Learning

- Naive Bayes:
  - Can be used for continuous (online) learning quite efficiently.
  - Needs strategies to adapt to changes.

We've seen examples of algorithms that can be used to **update** predictive models with new training examples.

… but still need some strategy to **adapt to changes**.

Why Are Strategies to Adapt to Changes Necessary?

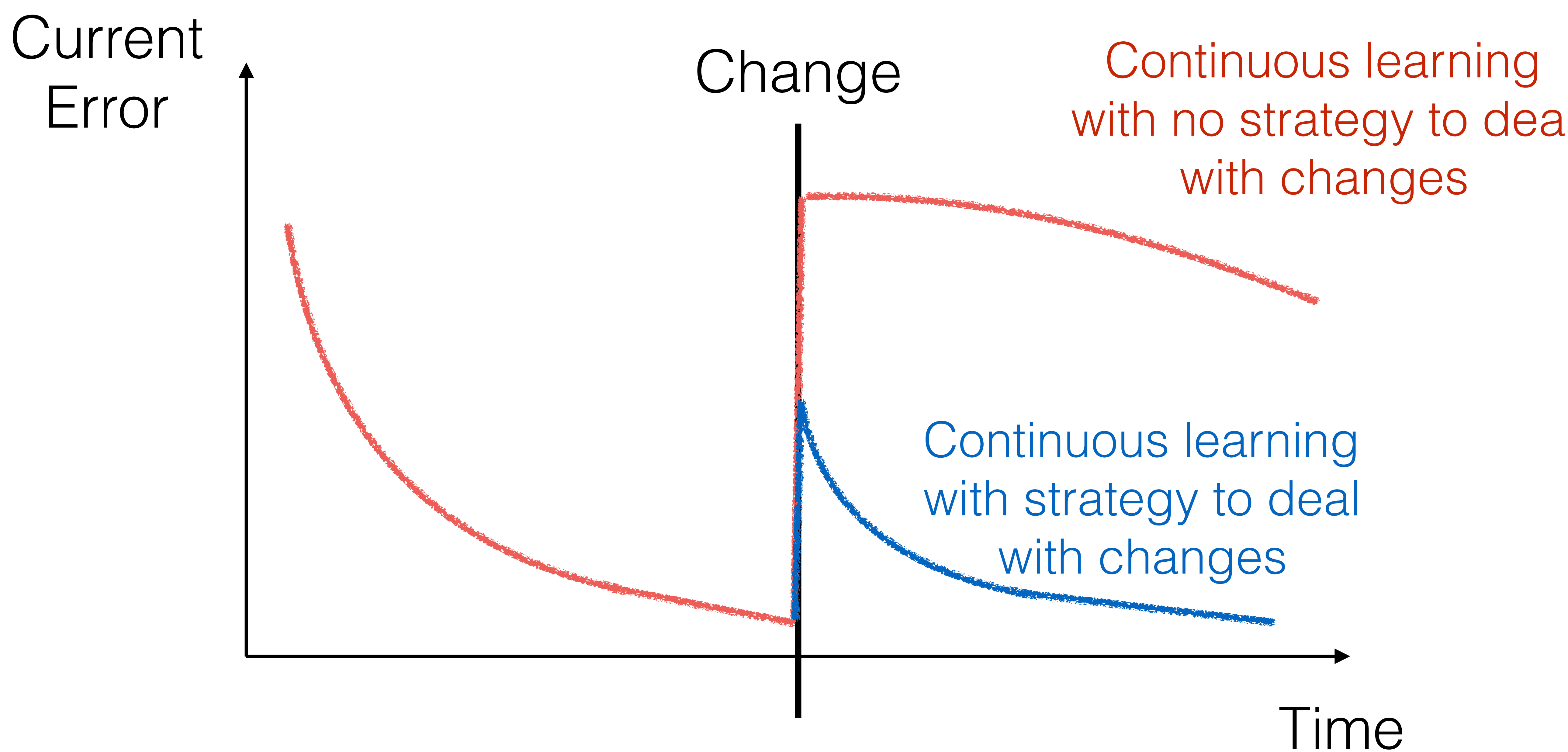
[YouTube video posted by : https://youtu.be/NXUUMLJbPCE]

Why Are Strategies to Adapt to Changes Necessary?

The coyote trained so much to catch the road runner, that now that he caught it he is struggling to adapt.

- A model is trained on new examples over time.
- When a change happens, many examples describing the situation before the change have been used.
- It would take many examples of the new situation to compensate for the old examples that were learnt.
  - Adaptation is slow.

Error Why Are Strategies to Adapt to Changes Necessary?



Dealing with Changes — Principles

Many approaches for dealing with changes are based on the following principles:

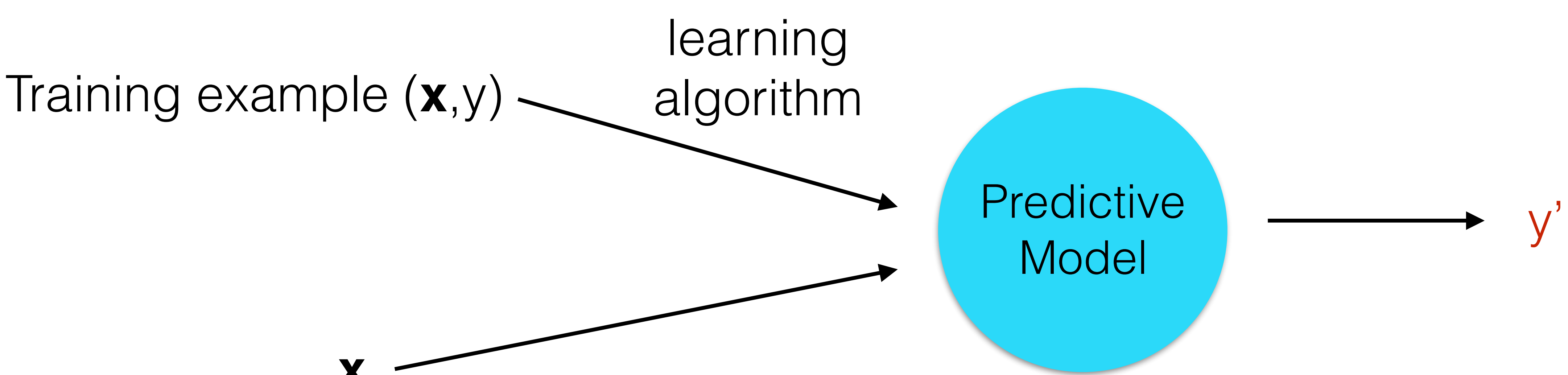
- Increase in error:
  - Error of a model is expected to reduce or stay constant over time if there is no change.
  - If the error of a model starts to increase, it is likely that there is a change causing this model to become inadequate.
- Building new model:
  - If current model(s) is(are) deemed inadequate for the current situation (increased error), create a new model to start learning the situation from scratch.

Ensembles for Adapting to Changes

- What should we do with old models if we detect a change?
  - Some approaches will throw them away.
  - However, this may waste useful knowledge in case the new situation is somewhat similar to the old one, or if a past situation reoccurs.
- Ensembles of learning machines could be used as a way to keep old predictive models while still being able to adapt to changes.
- New predictive models can be added to the ensemble if changes are suspected or detected.
- Weights could be used to emphasise the models that best reflect the current situation.
  - E.g., weight could be proportional to the accuracy of the predictive model since the change was detected.

Tracking a Model's Error Over Time

- Whenever a new training example arrives, we can use it to evaluate the predictive model before using it for training.
- This gives us an idea of how well the predictive model is performing as time passes.



Error on a single example

(y ≠ y') —> mistake

(y - y') —> error

Examples of Error Functions

- Classification error:
  - Given a data set D with examples ($\mathbf{x}_i$, y_i), $1 \leq i \leq n$.
  - The actual output (target) for $\mathbf{x}_i$ is y_i.
  - The prediction given by a classification model to $\mathbf{x}_i$ is y_i'.
  - y_i and y_i' are categorical values.

$$\text{Classification error} = \frac{1}{n}\sum_{i=1}^{n}(y_i \neq y_i')$$

Examples of Error Functions

- Mean Squared Error (MSE):
  - Given a data set D with examples $(\mathbf{x}_i, y_i)$, $1 \leq i \leq n$.
  - The actual output (target) for $\mathbf{x}_i$ is y_i.
  - The prediction given by a regression model to $\mathbf{x}_i$ is y_i'.
  - y_i and y_i' are numerical values.

$$\text{MSE} = \frac{1}{n} \sum_{i=1}^{n} (y_i - y_i')^2$$

First Problem of Error Functions

- These error functions need to know the errors on all examples in order to compute the error.

Classification error = $\frac{1}{n}\sum_{i=1}^{n}$ ($y_i \neq y_i'$) MSE = $\frac{1}{n}\sum_{i=1}^{n}$ $(y_i - y_i')^2$

- This would mean that we would need to store all these errors.
- If there is a lot of data, we will run into storage issues.
- But… we can re-write these equations to update the error whenever a new prediction is made.

Calculating Error in an Online Way

- The notation for the example iterator is typically *t* instead of *i*: ($\mathbf{x}_t$,y_t)
- Classification error:

$$\text{Classification error}_t = \begin{cases} (y_t \neq y_t') & \text{if } t=1 \\ \frac{t-1}{t} \text{ classification error}_{t-1} + \frac{1}{t}(y_t \neq y_t') & \text{if } t > 1 \end{cases}$$

Each example contributes equally to the error.

We have seen t examples so far.

Weight given to a single example: $\frac{1}{t}$

Weight given to t - 1 examples: $\frac{t-1}{t}$

Calculating Error in an Online Way

- Classification error:

$$\text{Classification error}_t = \begin{cases} (y_t \neq y_t') & \text{if } t=1 \\ \frac{t-1}{t}\ \text{classification error}_{t-1} + \frac{1}{t}(y_t \neq y_t') & \text{if } t > 1 \end{cases}$$

- Mean Squared Error (MSE):

$$\text{MSE}_t = \begin{cases} (y_t - y_t')^2 & \text{if } t=1 \\ \frac{t-1}{t}\ \text{MSE}_{t-1} + \frac{1}{t}(y_t - y_t')^2 & \text{if } t > 1 \end{cases}$$

Exercise

Calculate the classification error using the original classification error function and the online classification error function for the following examples and corresponding predictions:

- y_1 = false, y_1' = true
- y_2 = false, y_2' = false
- y_3 = true, y_3' = false

Second Problem of Error Functions

- Equal importance is given to all training examples.
  - Past examples are given the same importance as more recent ones.

- If we want to find out how well a model is doing currently, we need to modify the equations.

Exponential Decay Error Functions

- Classification error:

$$\text{Classification error}_t = \begin{cases} (y_t \neq y_t') & \text{if } t=1 \\ \dfrac{t-1}{t}\,\text{classification error}_{t-1} + \dfrac{1}{t}(y_t \neq y_t') & \text{if } t > 1 \end{cases}$$

- Mean Squared Error (MSE):

$$\text{MSE}_t = \begin{cases} (y_t - y_t')^2 & \text{if } t=1 \\ \dfrac{t-1}{t}\,\text{MSE}_{t-1} + \dfrac{1}{t}(y_t - y_t')^2 & \text{if } t > 1 \end{cases}$$

Exponential Decay Error Functions

- Classification error with exponential decay:

$$\text{Classification error}_t = \begin{cases} (y_t \neq y_t') & \text{if } t=1 \\ \eta \text{ classification error}_{t-1} + (1 - \eta)\,(y_t \neq y_t') & \text{if } t > 1 \end{cases}$$

- Mean Squared Error (MSE) with exponential decay:

$$MSE_t = \begin{cases} (y_t - y_t')^2 & \text{if } t=1 \\ \eta\ MSE_{t-1} + (1 - \eta)\,(y_t - y_t')^2 & \text{if } t > 1 \end{cases}$$

η, $0 < \eta < 1$, is a weight representing the importance of the past

Further Reading

Issues in Evaluation of Stream Learning Algorithms

João Gama, Raquel Sebastião, Pedro Pereira Rodrigues

Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining

Pages 329-338

http://readinglists.le.ac.uk/lists/D888DC7C-0042-C4A3-5673-2DF8E4DFE225.html

http://dl.acm.org/citation.cfm?id=1557060