Learn Tutorial

# Introduction

## Design

### Overview

The Learn framework is designed as a plug-and-play model for machine learning on streaming data. This type of abstraction enables a wide variety of machine learning algorithms to be used on streaming data. There are two main applications for which the Learn framework can be used: continual learning and batch learning.

#### Continual learning

Continual learning refers to problems in which machine learning is continuously run over a stream of data. For example, an algorithm for determining credit card fraud may use the last 200 transactions to determine if the current transaction is an anomaly. Over time, such an algorithm will adapt to the customer’s recent purchasing patterns and will continually update itself.

#### Batch learning

Batch learning refers to problems in which machine learning is run once on an offline dataset. This is usually the prevalent scenario for common machine learning problems. However, such problems usually deal with an offline dataset for training and an offline dataset for prediction. Batch learning as used with streaming data differs from this. A machine learning model is trained once for an offline dataset as in offline batch learning; however, prediction values are generated continuously for a stream of data. For example, in a scenario where the user wants to analyze the sentiment of Twitter tweets in real-time, a machine learning model may be trained once on a labeled dataset and applied in real-time to a stream of tweets.

### Goals

The Learn framework aims to facilitate the use of machine learning algorithms on streaming data. In this perspective, the goal is to minimize the amount of time a novice programmer needs to implement such an application. We define novice programmer as a person proficient in Python yet new to streaming data architecture. As the Learn framework is built on top of PSTREAMS, it is very easy for a user to write code that runs machine learning algorithms on streaming data without advanced knowledge of the streaming architecture.

## Machine Learning Models

Machine learning models are mathematical functions in a defined hypothesis space that are used to predict output values for input values. A machine learning algorithm has two components: training and predicting. Given a training dataset, the algorithm first trains a model. It then uses the model to generate predictions. To abstract this process, the Learn framework operates by splitting machine learning algorithms into these two components. The plug-and-play functionality comes from the Learn framework’s ability to accept user-defined functions that run training and prediction respectively. To use the framework effectively, the user needs to take the following steps:

1. Determine the type of function desired to learn.
2. Implement a training function that uses a training dataset to train the function.
3. Implement a prediction function that uses the trained hypothesis function to generate a prediction value for an input.

These steps are general to machine learning; it is important that the user be able to recognize and separate a machine learning algorithm into these specific components.

There are two main classes of machine learning: supervised learning and unsupervised learning. Supervised learning refers to scenarios where the training dataset has labeled outputs and the goal is to learn these outputs generally. For example, a user may want to predict the price of a house given its location, year of construction, amenities, etc. In this case, the training dataset would consist of data points for many houses with the price labeled. Unsupervised learning refers to scenarios where the training dataset does not have labeled outputs and the goal is to learn patterns in the data. For example, re-examining the example discussed earlier about credit card fraud detection, the training data does not have any labels regarding if a transaction was an anomaly or not. Instead, the machine learning algorithm attempts to learn the underlying patterns in the transaction history to determine if the current transaction is an anomaly.

The Learn framework is extremely flexible, supporting both supervised and unsupervised learning. The user only has to ensure that the training and prediction functions provided are consistent with each other.

# Basics

The Learn framework takes many parameters. These parameters are described in the documentation for the Stream\_Learn class. The Learn framework takes different parameters depending on if continual learning or batch learning is being used.

We go through a few examples to illustrate the use of these parameters.

## Continual learning

### Linear Regression

This example uses linear regression to predict trends for changing data. It is assumed that the user has basic knowledge of NumPy.

We first begin with importing the necessary modules to run the example.

INSERT CODE FOR IMPORTS

The Learn framework contains a class called Stream\_Learn, which runs the machine learning algorithms on the streaming data. The LinearRegression class contains training and prediction functions for using linear regression on streaming data. These functions are already prewritten and the user does not need to write new functions. The linear\_regression module contains helper functions for plotting the data as well as additional functions to run linear regression. We only need to use this module for plotting the data.

Next, we define some parameters.

INSERT CODE FOR PARAMETERS