A New Journey to Machine Learning in Quantitative Finance

Hao Ni
University College London
The Alan Turing Institute

Summary of the Course

Outline of the Course

- Week 1: Overview of ML in Quantitative Finance;
- Week 2: Supervised learning framework. Linear regression and regularization methods.
- Week 3: Deep neural network (DNN) and an application of DNN to derivative pricing;
- Week 4: Recurrent neural network (RNN) and an application of RNN to predict limit order book.

Supervised Learning Framework

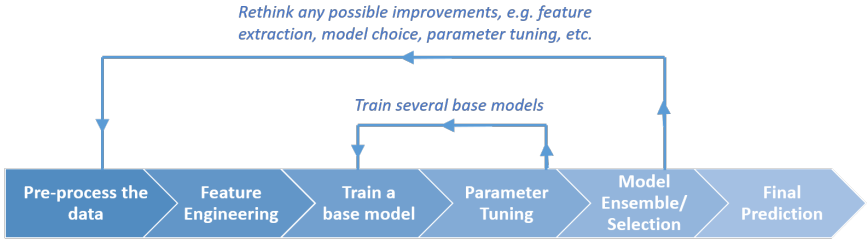
	Regression	Classification
Dataset:	$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^N, (x_i, y_i) \in \mathcal{X} \times \mathcal{Y}$	
	$\mathcal{Y} = \mathbb{R}^d$	$\mathcal{Y}$ is a finite set.
Model:	$f_\theta(x) (\approx \mathbb{E}[y\|x])$	$f_\theta(x) (\approx \mathbb{P}[y\|x])$
Empirical Loss:	$L(\theta\|\mathcal{D}) \rightarrow$Minimize	
	(e.g. MSE)	(e.g. Cross entropy)
Optimization:	$\theta^* = \arg\min_\theta (L(\theta\|\mathcal{D}))$	
Prediction:	$\hat{y}_* = f_{\theta^*}(x_*)$.	$\hat{y}_* = \arg\max_{i \in \mathcal{Y}} f^i_{\theta^*}(x_*)$.
Validation:	Compute the test metrics	
	e.g. MSE.	e.g. Accuracy.

Table: Summary of the framework of regression and classification. The difference between them is highlighted in blue.

Method	Linear regression (with regularization)	Neural Networks
Model	$f_\theta(x) := \theta x$	ANN, RNN, etc
Empirical Loss	MSE (+ penalty term of $\|\|\theta\|\|$)	MSE, etc
Optimization	Analytic formula for θ^*	Stochastic/Mini-batch Gradient descent
Prediction	$\hat{y}_* = \theta^* x$	$\hat{y}_* = f_{\theta^*}(x_*)$.
Validation	Compute test metrics (e.g., MSE, R^2)	Compute test metrics (e.g., MSE, R^2)

Table: Summary of linear models (with regularization) and neural networks for regression tasks.

ML pipeline to tackle financial data problems



The first step is to understand data and pre-process the data, which mainly includes data visualization, clean (remove outliers, missing values) and data normalization.

Select important factors and extract ad-hoc, application-dependent features.

Choose suitable supervised learning algorithms, e.g. linear models, gradient boosting trees, cnn, etc.

Find optimal hyper-parameters by cross validation and refine the model.

Build a better model using various base models or select the best one.

Make the prediction of the ensemble model in the testing data.

Thanks for your attention!