Applying Machine Learning and Predictive Analytics in FinTech

CISC 849 -- Applications of Financial Technology
September 8, 2015
Big Data in Financial Analytics

- Highly dimensional data
- Various domains of information
  - Web, finance, marketing, commerce, industry, security
- Heterogeneous – different data sources have different formats
- Traditional statistical methods are obsolete
Solving the Big Data Problem

• Machine Learning provides a novel approach toward processing, analyzing, and producing results from large datasets because:
  – They are generic
  – Machine learning provides an automated approach
  – Algorithms can be far more efficient
Types of Data

- Arrays/matrices of numbers
- Strings of characters (e.g. name)
- Graphs/Networks
- Functions/Curves/Time Series
Statistical Learning Milestones

- 1943 – Artificial neuron model
- 1958 – Perceptron algorithm
- 1971 – Uniform laws of large numbers
- 1974, 1986 – Backpropagation algorithm
- 1984 – CART
- 1995 – Statistical learning theory
How can we learn?

- **Approach**: Statistical Learning Theory
- **Problems**: Classification, Regression, Anomaly Detection, Ranking, Source Separation
- **Frameworks**: Supervised, Unsupervised, Semi-supervised, Reinforcement, Sequential, Iterative
- **Methods**: Decision Trees, Support Vector Machines, Boosting, Random Forests, Kernel Methods, Minimum Volume Sets
- **Concepts**: Risk Minimization, Complexity Regularization, Sparsity, Clustering
How can we implement ML solutions?

1. Define the type of problem
2. Choose an adequate/appropriate method
3. Get data in the correct format for the method
4. Construct a model
5. Validate the model
6. Resample data
7. Reiterate
Machine Learning and Predictive Analysis Tools

- C (libSVM)
- Python (SciKitLearn)
- MATLAB + SVM Toolbox
- R Project
- WEKA
How does Machine Learning and Predictive Analytics Work?

• Predictions based on past data
• Higher dimensionality of data provides additional learning “characteristics”
• Optimization methods with efficient impl.
• How the data is processed: batch, online, one-pass
UNSUPERVISED LEARNING
Indexing / Filtering Data

• Find **sparse representations** $P(X)$ of data $X$
• **Fast search** among a dictionary $D$
• Selection of a **loss function** (e.g. argmin distance)
• **Fast segmentation of financial return series**
  – Automatically identify (linear) trends and durations
  – Describe residuals: variance, skewness, etc
Applications

• Features produced by the learning algorithm can be used to:
  – Represent successive market regimes efficiently
    • This is done through the use of clustering
  – Detect market anomalies/opportunities
    • Minimum Volume set estimation vs VaR analysis
  – Feed a supervised learning algorithm
    • Prediction of (trends of) returns
    • Portfolio optimization
Summarizing Dependence Across Assets

• Conditional dependence structure can be represented in one of two ways:
  – Bayesian networks
  – Markov networks
SUPERVISED LEARNING
Predictive Learning

1. Data
2. Preprocessing
   - Descriptors
3. Descriptive tool
4. Predictive tool
   - Learning algorithm
   - Prediction rule
5. Monitoring tool
   - Post processing
   - Indicators
   - Alarms
6. Prediction
   - Confidence
   - Interpretation
Setup for supervised learning

- Random pair \( (X, Y) \sim P \) unknown
- \( X \) = observation vector in \( \mathcal{X}(\mathbb{R}^d) \)
- \( Y \) = label in \( \mathcal{Y} \subset \mathbb{R}^d \)
- Predictor: \( g : \mathcal{X} \rightarrow \mathcal{Y} \) in a class \( \mathcal{G} \)
- Loss function: \( \ell : \mathcal{Y} \times \mathcal{Y} \rightarrow \mathbb{R}^+ \)
- Risk functional (unknown!) = Generalization error
  \[
  L(g) = \mathbb{E} \left( \ell(Y, g(X)) \right)
  \]
- to minimize over \( g \in \mathcal{G} \).
Empirical Risk Minimization

- Data $D_n = \{(X_1, Y_1), \ldots, (X_n, Y_n)\}$
- Learner: $f : \mathcal{X} \to \mathbb{R}$ in a class $\mathcal{F}$
- Practical loss function: $\tilde{\ell} : \mathcal{Y} \times \mathcal{Y} \to \mathbb{R}^+$
  In general, convexified/penalized surrogate of $\ell$
- Empirical risk functional = Training error

$$R_n(f) = \frac{1}{n} \sum_{i=1}^{n} \tilde{\ell}(Y_i, f(X_i))$$

to minimize over $f \in \mathcal{F}$.
- Solution: $\hat{f}_n = \arg\min_{f \in \mathcal{F}} R_n(f)$
Overfitting
Issues

• Rates of convergence
  \[ L(\hat{g}_n) - \min_{g \in \mathcal{G}} L(g) \leq C(n, \mathcal{G})? \]

• Empirical estimation of \( L(\hat{g}_n) \) and validation
  \[ \rightarrow \text{Cross-validation, leave-one-out, bootstrap estimates} \]

• Interpretation of prediction rules

• Sparse representations

• Online vs. batch learning, parallelization

• Multitask prediction
Example Process (PRISMS)

1. **Market data**
   - Several asset classes (FI, FX, commodities, etc.)
   - Relevant information is extracted (filtering)
   - Each predictor is characterised by its momentum and volatility

2. **Learning**
   - Learn from market data without assumptions
   - Analyse market configurations
   - Monitor large datasets

3. **Risk monitoring**
   - Characterize the current market configuration
   - Rank market drivers
   - Produce global risk indicators

4. **Recommendations**
   - Predict trends and reversals (absolute or relative)
   - Return confidence indicators for each signal
STATISTICAL LEARNING
Statistical Learning Overview

- **Learn from data:**
  - Analyse great amounts of data.
  - Infer dependence structures from observations.
  - Detect patterns in massive databases.
  - Focus on prediction accuracy and optimization of statistical performance.

- **Types of tasks:**
  - Supervised: considering known labels associated to data, one wants to learn the relationship between observations and labels.
  - Unsupervised: one wants to assign labels to data elements by clustering them into consistent groups.

- **Examples:**
  - Medical diagnosis from clinical data, object recognition in images, etc.
  - Speech separation, homogeneous regions clustering, social network peer groups, etc.
Statistical Learning Problems

Problems can be formulated as learning tasks:

- **Market prices representation as successive periods of homogeneous trend is a typical unsupervised problem.**
  - Segmenting a single signal into regimes implies clustering dates into homogeneous groups.
  - Market regime identification assigns to each period a cluster label, the regime it belongs to.
  - Such indexing provides a market description.

- **Predicted return is a challenging supervised problem.**
  - One is interested in a few categories of asset return (up/down, possibly strong/weak trend).
  - Given an observation of the market, one wants to classify the future return into one of the categories.
  - Based on past observations of the dependence between the market and the future return, a classifier model can be learnt.

- **Risk factor analysis is a matter of source separation.**
  - Analyzing a portfolio behaviour relatively to the market involves breaking it down to predominant factors.
  - Those factors must be independent, as if the portfolio value originated from distinct signal sources.
  - Statistical learning provides robust techniques to perform such a task.
Statistical Learning Outcomes

Ability to analyse high dimensional data
- Market history is an overwhelmingly large database containing information.
- Learning algorithms can handle problems with highly multivariate behaviours.
- Underlying information is automatically analysed and used.

Early arbitrage opportunities
- We believe there are inefficiencies in the market and that mean reversion is not instantaneous.
- A model that efficiently learnt from historical data can reveal these phenomena.
- Such a tool is likely to detect arbitrage opportunities before other observers.

Rational information for decision making
- Data-driven algorithms bear the minimum hypotheses possible, making them fully objective.
- Automatically learnt models are complementary to other models and heuristics.
- Human decision making is enhanced.
Statistical Learning Methods (1/3)

Signal indexing representation

- Minimization criterion: least square regression error
- Dyadic Coifman-Wickerhauser (bottom-up)
  - Initial state: consider all dates separated
  - Merge consecutive pairs of dates in case the criterion decreases
  - Iterate until the criterion cannot be improved or the maximum merging degree is reached
- Unsupervised CART (Classification and Regression Trees, top-down)
  - Initialisation: consider the whole history as one period
  - Find the best splitting date so as to minimise the criterion
  - Iterate until the criterion cannot be improved or the maximum splitting degree is reached

Market regime identification

- K-Means:
  - Minimise intra-cluster distances
  - Iteratively find cluster “centroids” and data labels
Classification with Random Forests

- Base classifier: decision trees (CART)
  - Objective: find an optimal partition of the input space, defined by a tree.
  - Criterion to minimise: impurity, known as the Gini discriminant index, measuring how mixed a set is.
- Randomization and aggregation
  - In order to increase robustness, several subsets of data are sampled.
  - At each node, the best splitting variable is chosen among a random sub-sample of all the variables.
  - Build trees on the subsets and make them vote.

Extension to several trend levels

- The algorithm is intrinsically designed for any number of classes.
- Pre-processing is necessary to define levels such as strong/weak or downtrend/uptrend.
**Backtesting**
- The method is tested *with past events*
  - On each backtest date, a model with the same parameters is built and tested on following days.
  - Performance is assessed by backtest date and number of consecutive test days.
- Predominant factors are identified.
  - Each model selects the important variables.
  - Variable ranks evolve in time following the economic environment.
- Parameters are calibrated to maximise expected performance.

**Monitoring**
- Predictions are always associated to confidence measures such as:
  - similarity to training data,
  - local known performance in the neighbourhood of the new observation.
- Those measures can provide alerts on the model's reliability.
Statistical Learning Inputs

Possible predictors:

- Economic variables: currencies (4), FI (25), commodities (5), indices (30).
- Stocks (index constituents).
- In the near future, financial variables such as P/E, dividend yields and EV/EBITDA.

Variables to predict:

- Stock prices (trends).
- Index, commodity, currency and FI prices.
- Potentially, volumes, volatilities, and correlations.
Statistical Learning Outputs

Analysis
- Data indexing
- Market regime mapping

Prediction
- Predicted directions with confidence indicators
- Market scenarios
- Assets and predictors ranking

Backtesting and monitoring
- Monitoring alerts
- Strategy backtests
Statistical Learning Task Complexity

**Input space dimension:**
- Input variables: 128 dimensions
  - 64 predictors
  - 2 dimensions per predictor (last trend and volatility)
- Usual training data: 250 dates
  - 2-year initial daily data
  - 1-year sliding window market representation

**Usual computing time:**
- Standard model learning (5 minutes*)
  - Indexing a usual set of predictors: 10 minutes* at the first run, 4 minutes* for subsequent use
  - Training models for 5 classes: 6 seconds* per model
  - Prediction: < 1 second per predicted value
- Backtesting
  - Learning and testing on 3-year input data
  - 250 models trained and tested: 20 minutes*
Risk Analysis and Portfolio Optimization

1. Systematic space
2. Explanatory variables
3. Risk breakdown and Betas