ML Review

Machine Learning for Trading (CS 7641)

- example x's: price, bollinger, momentum; y's: return and future price
- Supervised regression learning: linear regression, KNN, trees, forests
- RL
    - Markov decision problem: set of states (in market), set of actions (buy, sell, nothing), transition function T(state, action, state), reward function R(state, action)
    - Q Learning: Q'[s,a] = (1-alpha) * Q[s,a] + alpha * improved_estimate
- Induction, deduction, supervised learning, unsupervised learning (news, genes), RL, decision trees
- Definitions: instances (input), concept (function), target concept (answer), hypothesis (set of all possible concepts), sample (training set), candidate (possible target concept), testing set. Tree nodes = attributes, edges = values
- Preprocessing (clean, train/test) -> Learning (model, cross-validate, performance, params optimization) -> Evaluation (against test) -> Prediction
- Target function: type of knowledge to be learned (e.g. each possible checkers board score)
- Representation for target function (e.g. linear function of board artifacts, # of pieces, # of pieces threatened)
- Learning mechanism: gradient descent
- Decision Tree: Robust to errors and missing attributes
- Entropy: sum(-p*logp)
- Overfit -> Occam's razor
- NN: Well suited for noisy, complex sensor data, such as inputs from cameras and microphones
- Gradient descent derivation
- Sigmoid: 1/(1-e^-y)
- Instance based learning (KNN): training data in storage and use it to make a forecast
- Other algorithms reduce data into a function, then make predictions ignoring the data
- Bayesian Learning: P(h | D) = P(D | h) * P(h) / P(D)

$$P(disease|pos) = 21\% = 98\% * 0.8\% / (98\% * 0.8\% + 3\% * 99.2\%)$$

- Pros/cons: DT, KNN, Boosting, SVM, NN, Bayes, Random Opt/Annealing
- Review
  - Maximum a posteriori MAP = argmaxh P(D | h) * P(h)
  - Maximum likelihood ML = argmaxh P(D | h)
- EM: Will not diverge, may not converge, can get stuck - must random restart
- K-means: randomly select k centers (points), assign each point to cluster (E), recalculate centers (M)
- Feature selection: exponential problem 2^n
- Filtering: selecting features first, then running algo, ignores learning problem, fast. Can use decision tree to select which features give most info. gain, then pass those features to another learner such NN
- Wrapping (optimizing algorithm based on given features and reiterating, slower). forward search (run learner with each feature independently, keep the best, add one more feature in addition to first and iterate as long as error improves)
- Feature transformation: PCA (Maximizes variance, Mutually orthogonal), ICA (Cocktail party problem: extract voice from multiple sources in a party)