MACHINE LEARNING
BARUCH COLLEGE
SPRING 2015
T.A.: LIQUN ZHU

Homework Assignment 5

1. You're searching for Association Rules in a large transaction database with about 100 million records. You've decided to *partition* your database. You do not wish your Minterm Support error to exceed one tenth of 1%, with a 99% confidence. Ignoring the bias introduced by not sampling with replacement, how many database partitions should you examine? As always, you want the global Minterm Support to stay around 5 in order for your χ^2 statistic to be robust. Given the size of your partitions, would you have to reduce the Support threshold significantly when searching for Supported Itemsets within the partitions?

2. To construct an XOR classifier with two inputs, would you use a Radial Basis Function Neural Network or a Feed-Forward one? Why? Given that the XOR problem has *Maximal Linear Complexity*, would your answer change if you needed an XOR classifier with a 10-dimensional input space? Why or why not? Would you use the input dimension or the *Linear Separability Index* to decide between using a RBF-NN vs. a FNN? Discuss.

3. Can you suggest a way to use an *Auto-Associative* FNN to approximate the *Linear Separability Index* of an (input) space? Would this alleviate the *Curse of Dimensionality*? Can you suggest a way that an Auto-Associative FNN can be used in conjunction with an RBF-NN to improve overall performance? Discuss.

4. Given two arbitrary Perceptrons solving the same Linearly-Separable problem, show that the Perceptron with least-energy weights will generally have better classification performance. If you are using an FNN instead of a Perceptron as a classifier, can you suggest how to modify the penalty (error) function to train the FNN to produce a higher-performing classifier?

5. Given a noisy return series Y_i^*, you train a model to extract a prediction $\hat{Y}_i$. You improve this prediction further by bootstrapping many models to produce an improved prediction $\hat{Y}_i^B$ consisting of the average of the individual predictions $\hat{Y}_i$. You can think of the original noisy signal as consisting of the bootstrapped prediction corrupted with additive noise: $Y_i^* = \hat{Y}_i^B + \varepsilon_i$, with $\varepsilon_i \sim \mathcal{N}(0, n\sigma_Y)$, where

σ_Y is the standard deviation of the (unobservable) uncorrupted (predictable) time series Y_i which you estimated as $\hat{Y}_i$. The bootstrap estimate itself can be thought of as the individual estimates $\hat{Y}_i$ corrupted by the "bootstrap noise" coming from the dispersion in the individual bootstrap predictions: $\hat{Y}_i^B = \hat{Y}_i + \varepsilon_i^B$, with $\varepsilon_i^B \sim \mathcal{N}(0, \sigma_B)$. Here σ_B is the standard deviation of the individual bootstrap predictions $\hat{Y}_i$, which we can always write as: $\sigma_B \equiv n_B \sigma_Y$, where usually $n_B \ll n$. Note that, while σ_B is observable by taking the standard deviation of the individual model predictions, σ_Y is not directly observable. We write it in this way so that the unobservable σ_Y will drop out of our calculations, and we will shift the unobservable quantity to n_B. You now assume that the expected return of your trading strategy is proportional to the volatility of the bootstrapped prediction, σ_{Y^B}: $r = \lambda \sigma_{Y^B}$, while the risk of the strategy is the volatility of the corrupted series Y_i^*. Assuming that the bootstrap error ε_i^B is uncorrelated with the corrupted signal error ε_i, (a) what is the Sharpe Ratio of your strategy in terms of λ, n, and n_B? (b) By how much have your R^2 and SR increased by performing the bootstrap? (Compare your answer above to Problem 1 in HW#2 and Problem 1 in HW#3; you may exploit the fact that $n_B \ll n$ to simplify your answer.)

6. Draw a diagram of a Recurrent FNN that would implement an ARMA(*p,q*) time-series prediction where p and q are the Autoregressive and Moving-Average orders, respectively, showing appropriate TDLs (if needed), and recursion (if needed). Explain all notation used, and describe how the network implements an ARMA(*p,q*) model. Why is Bootstrapping problematic when used with Time Series models?