Forecasting:
Preliminary Design Review

By Brieanna Sundberg, Grant Liu,
and Tyrin Besas

Contents

- Project Overview
- Progress Update
- Plan of Attack
- Upcoming Plans
- Problems Encountered
- Potential Problems
- Questions

Project Overview

- The forecasting team focuses on analyzing data and using machine learning and other prediction techniques to track trends.
- Beginning with raw data, we pre-process it so that the code is able to be passed through various predictive algorithms.
- The idea behind weather prediction is that we will one day be able to predict changes in the solar energy produced, and can modify the amount of power the grid draws to maintain a balance of power.

Progress Update

- Reviewed Fall 2016 Code
- Regression Example with Seyyed
- Introduction to Recursive Least Squares
- Began Documentation of Algorithms & Code with LaTeX

Least Squares Example

Air Quality Least Squares Example:

Using an air quality data set obtained from the UC Irvine Data Repository we were to create models predicting the concentration of CO in the air.

From this example we learned many things, including:

- Practice with coding in Python
- Practice with Plotting data in Python
- Practice with the Least Squares Algorithm
- A better understanding of Feature Extraction

Least Squares Example Cont.

The Least Squares Algorithm works by determining a weights vector that minimizes error. Using the input data, X matrix, and the desired output, D vector, we can determine the weights using the following equation. $(XX^T)^{-1}XD$ This simplifies to $(X^T)^{\dagger}D$ using the pseudoinverse.

Process:

1. Delete Erroneous Data
2. Construct X Matrix and D Vector
3. Find Weights Vector
4. Create Prediction
5. Calculate the Error

Least Squares Example Cont.

By carefully selecting different features to include in our models we can gain different insights about the nature of the data.

- Find CO concentrations as a function of:
  1. Time
  2. Temperature, Relative Humidity, Absolute Humidity, and the Sensor values
  3. Temperature, Relative Humidity, Absolute Humidity
  4. Sensor Values

Findings from RMSE (Root Mean Squared Error):

1. RMSE = 1.5457
2. RMSE = 0.6662
3. RMSE = 1.4304
4. RMSE = 0.6841

$$\text{RMSE} = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(Y_i - \hat{Y}_i\right)^2}$$

Online vs Offline

- Offline Algorithm
  - An entire data set is input at once.
  - Model is constructed off of this, and is applied to other data sets
- Online Algorithm
  - Model is updated continuously as new data comes in
  - Each time a new data point is entered, weights are recalculated to better fit the data set, constantly adjusting to the changes in the data

Recursive Least Squares

- Online version of Least Squares
  - Continuously takes in new data and updates the model.
  - Uses Woodbury matrix identity to remove the inverse operation which is the most computationally complex.
    - Using this technique you update a series of variables that together determine the updated weights

Plan of Attack

- Meet weekly with Seyyed to learn new concepts & techniques
- Create documentation for algorithms and concepts to leave for future forecasting teams
- Implement concepts in Python
- Create robust functions that can be easily understood and utilized

Up Next

- Continue learning about Recursive Least Squares
- Implement an example
- Implement Recursive Least Squares into a larger solar irradiance code
- Continue work on documentation

Possible Upcoming Plans

Machine Learning Algorithms:

Least Mean Squares

Feature Extraction

Offline/Online

Supervised/Unsupervised

Linear/Nonlinear

Code & Documentation:

Increase familiarity with Python

Document Functions & Code

Document Concepts

Create Robust Working Code

Problems Encountered

- Learning new syntax and techniques
- Errors in collected data
- Transitioning conceptual algorithms into robust code
- Debugging code

Potential Problems

- Transitioning conceptual algorithms into robust code
- Debugging code
- Collaborative Code
- Documentation format

Questions?