Problem Statement

Build a system for gathering and correlating distributed inputs of noisy data. The system must form predictive models from clusters of inputs and correlations between them. The system must provide users with location-specific conditions based on these predictions and correlations. Users must be able to customize conditions or trends that trigger alerts and notifications, and transfer notifications to other users to allow for collaborative interaction.

Motivation

The foremost goal of the project is to provide the user base access to a sophisticated weather forecasting service that can:

1. Register user interest in particular geographical areas and weather conditions.
2. Notify the user of times in the future when a registered point of interest will likely satisfy their desired weather conditions.

However, the developed data retrieval and analysis system should not be constrained to just weather data. The intention is to build enough flexibility into the system that it can generate predictive models for any situation given the appropriate data. For instance, the system’s existing retrieval and processing tools could predict part failures in an oil rig when pointed at the appropriate data sources.

System Overview

At a high-level, our system will gather past data (oceanic, weather, etc.) to build models which will be used to predict future conditions by examining the current conditions.

A user will be able to interact with our system in a number of ways. A user can specify alarms for certain events such that the system will alert them when that event occurs. For instance, a user could set an alarm to inform them when water temperature is predicted to be above 75 degrees.

The user can also interact with the models and data more closely; the system can display its current predictions for all data which our models allow for. It can display representations of its models, past data, and current real-time data.

Finally, it is vital that our system be highly extensible, flexible, and automated. We plan to create a system which can be generalized to other problems, the ability to easily modify or add data sources, processes, models, and forms of user interaction is essential to our ultimate goal.
1. Connect to and gather input from multiple, simultaneous, time-sensitive, error-prone, quantitative data sources (system is not tied to just NOAA data).

2. Using collected data, the system needs to build statistical models with which can predict future conditions.

3. The system’s models must be able incorporate newly retrieved data to make more informed predictions.

4. The data can be noisy or erroneous, but the system must be able to:
   a. Filter noise.
   b. Detect and respond to outliers.
   c. Qualify predictions with a range of uncertainty.

5. The system needs to present the results of the correlation between the models and the recent data in a way that helps the user make a more informed decision. It should take the users’ preferences into account when providing output.

6. The system should support multiple clients and should be flexible to allow for future clients to be implemented on multiple platforms (iOS, Android, etc.).

7. Users should be able to receive notifications for specific locations when predicted conditions line up with their preferences.

8. The system should support multiple views. The casual user should not have the same level of control and options available as an administrator should. A mobile user should not receive the same level of information as a desktop user.

9. Users should be able to share their context (e.g: the measures they are looking at for a particular set of buoys) with other users to enable collaboration.

Requirements

1. Must demonstrate system capabilities using:
   a. Real-time buoy data
   b. Historical buoy data
   c. Local weather conditions
   d. Astronomy data
   e. Elevation
   f. Tidal data

2. Must make use of the Microsoft stack (.NET 4.5, WCF).

3. Must run on Azure.

4. Must conform to Schlumberger’s security standards.

5. Must be able to respond to data updates in a fraction of the time until the next update. In other words, if buoy data publishes every hour, the system must take a fraction of an hour to retrieve the update and update its predictions.

6. Must allow the user responsive interactions with the system -- response time must be at most a few seconds.

7. Models and user preferences must remain persistent.
8. Must not require the storage of retrieved data for system functionality.