Bayesian Network with Independent Random Variables

Karthik Bangalore Mani
Department of Computer Science
Illinois Institute of Technology

March 19, 2016

Abstract

The goal of this phase is to fit a Gaussian Model over the train data, and compute the parameters: $N(\mu,\sigma^2)$. The mean μ will be used to predict the values of humidity and temperature of the sensors, and the variance σ^2 will be used in determining the readings to be predicted during Active Inference phase.

Problem Statement

Given humidity and temperature Training datasets, estimate the Gaussian parameters, and check for the accuracy of predictions by computing the mean absolute errors. The mean absolute error is given by the equation:

$$\mathrm{MAE} = \frac{1}{n}\sum_{j=1}^{n}|y_j - \hat{y}_j|$$

Where n is the total number of readings, Yj is the actual reading, $\hat{Y}j$ is the predicted reading.
The prediction/ Inference is done using 2 techniques:

1. **Window Sliding:** If the budget is 15, then, the following sensors at the below time slots will communicate with the server , which in turn means, the MAE is zero for these sensors.

Time	Sensor
t	1 – 15
t + 1	16 – 30
t + 2	31 – 45
t + 3	46 – 50 ; 1 – 10

2. **Variance:** If the budget is 15, then, then at all time-slots, the sensors with top 15 variances will communicate with the server, which in turn means, the MAE is zero for these sensors.

Implementation Details

Function	Description
`get_data_matrix(file_path)`	Strip out the first row and first column and return data matrix
`get_reading_for_diff_days(data_matrix,train_days)`	Consolidate the readings of sensors across differe d days into a single tuple.

`get_params(consolidated_readings)`	Estimate Gaussian parameters : N(μ,σ)
`chunks(_list,size)`	Get Chunks of size size from a list _list
`get_windows(no_of_sensors,no_of_cols,budget)`	Get the window indices for each column/each time-stamp
`build_pred_mat_using_sliding(test_matrix,budget,modelled_matrix,max_cols=48)`	Predict the ones that are not in window. For the rest, use existing data
`mean_abs_error(actual_mat,pred_mat)`	Compute the mean absolute error
`top_k_indices(_list,k)`	Return the indices of top k elements of the list
`get_top_var_indices(mod_mat,budget)`	Extract the indices of cells with top k budgets
`build_pred_mat_using_Variance(test_matrix,budget,modelled_matrix,max_cols=48)`	Predict the ones that are not among the top budget variances. For the rest, use existing data
`predict(Data_sets,Inf_techniques):`	Predict the Readings

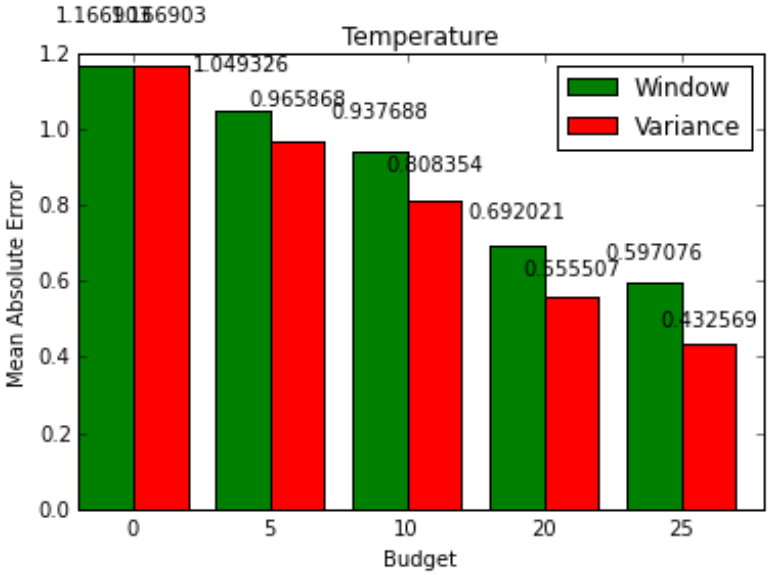
`get_ax(g_bar,r_bar,budgets)`
`autolabel(rects,ax)`
`plot(budgets,predictions)`
	Functions that help in plotting bar charts.

Results and discussion:

1. **Temperature Dataset :**
    a. For the temperature Dataset, **Variance** Active Inference was the winner as it had least MAE.
    b. Upon increasing the budgets, the MAE was observed to decrease.
    c. The budget and corresponding MAE were found to be :

Budget	Window MAE	Variance MAE
0	1.16690277778	1.16690277778
5	1.04932638889	0.965868055556
10	0.9376875	0.808354166667
20	0.692020833333	0.555506944444
25	0.597076388889	0.432569444444

d. The Bar Graph is :

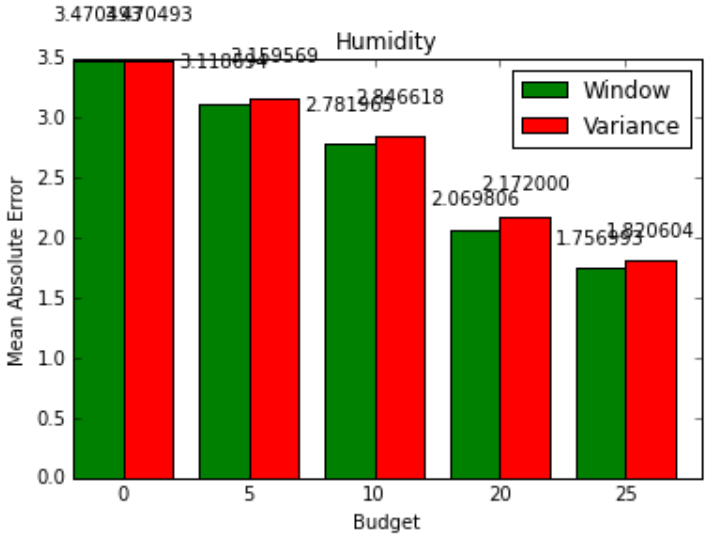


2. Humidity Dataset :

a. For the humidity Dataset, surprisingly, **Window** Sliding Active Inference was the winner.
b. Upon increasing the budgets, the MAE was observed to decrease
c. The budget and corresponding MAE were found to be :

Budget	Window MAE	Variance MAE
0	3.47049305556	3.47049305556
5	3.11869444444	3.15956944444
10	2.78196527778	2.84661805556
20	2.06980555556	2.172
25	1.75699305556	1.82060416667

d. The Bar Graph is : (***Next Page***)



Instructions to run program:

- Place datasets in the folder\Dataset\
- Run the iPython code
- View Bar Graphs in iPython notebook
- View result csv files in \Results\Temperature\ and \Results\Humidity\

References:

1. Stackoverflow.com
2. Wikipedia.com
3. Mathplotlib.org