

Mini Project 2

ECE 313

Probability with Engineering Applications

Lecture 13

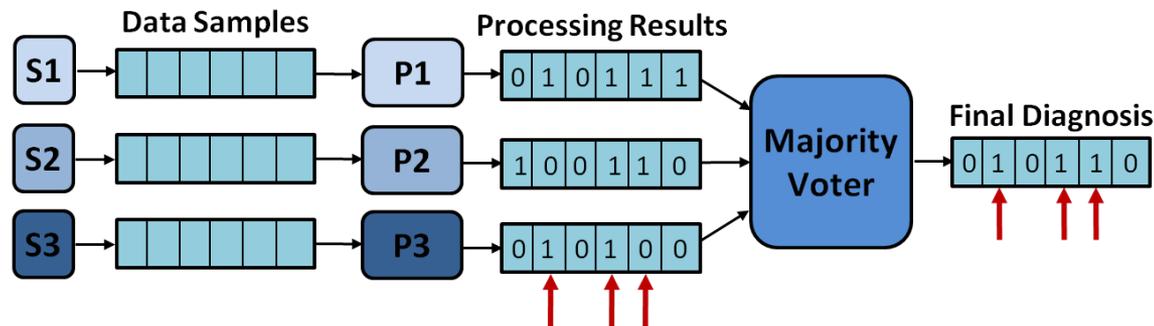
Professor Ravi K. Iyer

Dept. of Electrical and Computer Engineering

University of Illinois at Urbana Champaign

Mini Project 2: Description

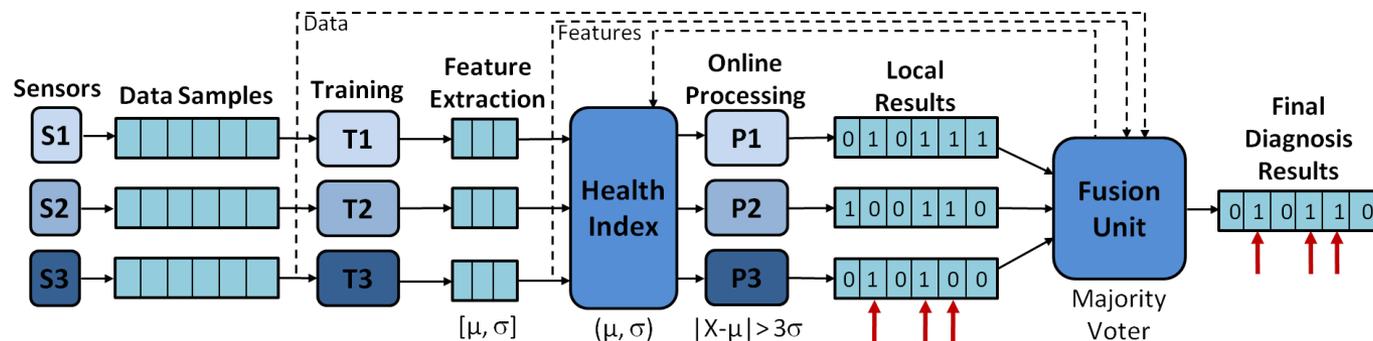
- Assume that you collect three correlated physiological signals from the biomedical sensors attached to the patient's body:
 - S1: Heart Rate (HR)
 - S2: Pulse Rate (PR)
 - S3: Respiration Rate (RESP)



- These signals are then passed through three processing units (P1, P2, and P3, respectively), called threshold functions, to detect patient abnormalities.
- Each threshold function will generate an alarm whenever a data sample of the corresponding signal exceeds a pre-defined threshold. A “1” on the output of each function indicates an alarm and a “0” corresponds to absence of an alarm.
- A majority voter function then generates the final output, based on the value that the majority of the threshold functions agreed upon.

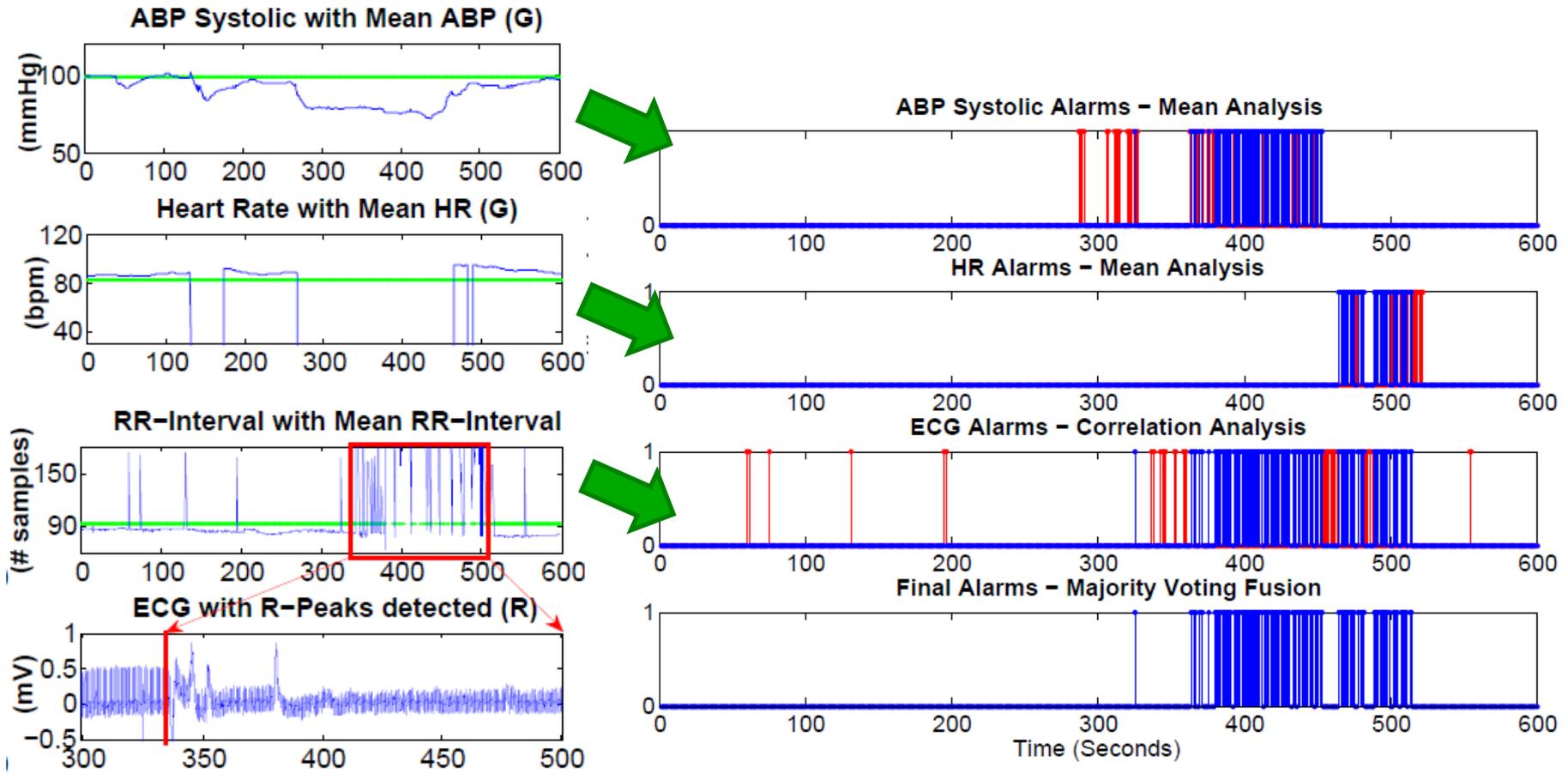
General Monitoring Flow

- The monitoring process starts with collecting multi-parameter inter-correlated physiological signals from biomedical sensors.
- Then it steps through an initial training phase, in which a physiological signature of the patient (Health Index), is compiled by aggregating different statistical features (mean, standard deviation, etc) from the input signals.
- During the monitoring phase, the obtained signature is used as a reference point (patient-specific threshold) for detecting abnormalities in each signal. At the end, a fusion technique (e.g. a majority voter) is employed to reach at a final diagnostic decision.



H. Alemzadeh, C. D. Martino, Z. Jin, Z. Kalbarczyk, R. K. Iyer, "Towards Resiliency in Embedded Medical Monitoring Devices," *Proceedings of DSN Workshop on Open Resilient Human-aware Cyber-physical Systems (WORCS-2012)*, Boston, MA, July 2012.

Majority Voter Example



Mini Project 2: Voter Success Examples

- Assume that by talking to the physician, we know that in the time interval T the patient experienced an abnormality and an alarm should be raised (The output from the monitoring system should be “1”).
 - If P1 and P2 functions generate “1”, but P3 generates a “0”, then the majority voter will generate a “1” (an alarm) on the final output, indicating a patient abnormality.
 - This means that P3 missed a *true alarm*, and voter *fixed a missed detection* on P3.
- Assume that for another time interval we know the patient experienced no abnormality and no alarms should be raised (The output should be “0”).
 - If P3 generates a “1”, but P2 and P3 functions generate “0”, then the majority voter will generate a “0” (no alarm) on the final output, indicating no patient abnormality.
 - This means that P3 raised a *false alarm*, and the voter *masked a false alarm* on P3.

Mini Project 2: Voter Failure Examples

- Assume that by talking to the physician, we know that in the time interval T the patient experienced an abnormality and an alarm should be raised (The output from the monitoring system should be “1”).
 - But both P1 and P2 functions generate “0”, while P3 generates a “1”, then the majority voter will generate a “0” (no alarm) on the final output, wrongly indicating that there is no patient abnormality.
 - This means that P3 raised a *true alarm*, but because P1 and P2 had *missed detection*, the voter output was erroneous. We call this a *missed detection*.
- Now assume that for another time interval we know the patient experienced no abnormality and no alarms should be raised (The output should be “0”).
 - If P3 generates a “0”, but P2 and P3 functions generate “1”, then the majority voter will generate a “1” (an alarm) on the final output, indicating a patient abnormality.
 - This means that P1 raise a *true alarm*, P2 and P3 raised *false alarms*, and the voter output was erroneous. We call this a *a false alarm*.

Mini Project 2: Task 0

- **Task 0:**
- Download and install MATLAB. You can get the student license from the UIUC WebStore (<https://webstore.illinois.edu/>) using your NetID
- You can also use R. Similar functions are available in R.
- You are provided with a dataset *patient_data.mat* and 3 functions *P1*, *P2*, *P3*
- Load the provided data set into MATLAB: `load patient_data.mat`
- *patient_data* consists of 2 variables: *data* and *golden_alarms*,
- *data* is an array of 3 rows and 30,000 columns
 - Each of the rows 1 to 3 respectively correspond to one of the signals:
 - Heart Rate (HR), Pulse Rate (PR), Respiration Rate (RESP)
 - Each column corresponds to a data sample
- **Hint:** Sample rate is 1 sample/second and the whole period is 30,000 seconds.
- *golden_alarms* is a binary vector of 3,000 columns
 - Each column corresponding to a 10-sample interval.

Mini Project 2: Task 1

- **Task 1:**

- a) Plot the Heart Rate (HR), Pulse (PR), and Respiration (RESP) signals over time.
- b) Use the *hist* function in MATLAB to plot the distribution of HR, PR, and RESP variables (**Hint:** Use `nbins = 50`).
- c) Calculate the mean and standard deviation of HR, PR, and RESP variables over the whole period.
- d) Use the *normrnd* function in MATLAB to generate three normally distributed random variables with the mean and variance that you found for each variable in part (c), then plot the distribution of these normal random variables. Do you see any difference between the distributions in part (b) and part (d)?
- e) Use the *normplot* function in MATLAB to estimate the difference between the each of distributions of signals in part (b) and their corresponding normal distributions.

Provide your explanations on the results in as a list of at least 3 bullets.

Mini Project 2: Task 2.1

- **Task 2.1:**
 - a) Use the provided threshold functions to generate alarms for each of the HR, PR, and RESP variables. Remember that a “1” on the output from each threshold function indicates an alarm and “0” corresponds to absence of an alarm.
 - b) Coalescing of alarms: The alarms generated for each of the variables HR, PR, and RESP are then coalesced separately over fixed intervals of 10 samples. For example, if for HR there is at least one alarm generated during an interval of 10 samples, then the output for that interval will be “1”, similarly for PR, and RESP.
 - c) Write a majority voter function that takes as input each of the 10-sample windows and votes on the alarms generated from each threshold function in part (b). The output of the voter will then a sequence of alarms that the majority (at least 2 out of 3) of the functions agreed on.
 - d) Use the *bar* function in MATLAB to plot the coalesced alarms generated for each variable (in part b) and the final alarms generated by the majority voter (in part c). (**Hint:** Each bar will represent an alarm (“1”) during a 10-sample window).

Mini Project 2: Task 2.2

- **Task 2.2:**

- You are given by the physician a vector of golden alarms (*golden_alarms* variable) which indicates for each 10-sample interval whether patient was diagnosed an actual abnormality (“1”) or not (“0”).

- A **false alarm** corresponds to the case where an alarm (“1”) is generated based on processed input data for an interval, but no abnormalities (“0” in the golden alarms) are detected by the doctor for the same input data.

- A **miss detection** corresponds to the case where no alarm (“0”) is generated based on processed input data for an interval, but at least one abnormality (“1” in the golden alarms) is detected by the doctor for the same input data.

a) Calculate the conditional probabilities of false alarms and miss-detections for the whole system (Show all your work).

- Recall that the voter operates on coalesced alarms, generated in part (b).

- $P(\text{False Alarm}) = P(\text{Voter raises an alarm} \mid \text{Physician indicates no abnormality})$

- $P(\text{Miss Detection}) = P(\text{Voter raises no alarms} \mid \text{Physician indicates an abnormality})$

b) Calculate the probability of error.

- $P(\text{Error}) = P(\text{Voter raises an alarm AND Physician indicates no abnormality}) + P(\text{Voter raises no alarm AND Physician indicates an abnormality})$

Project Timeline and Grading

- **A TA session will be held on Friday, 1:00 PM, in CSL 249, to answer any questions that you may have.**
- **A report (including both the MATLAB code, plots, and results) must be delivered electronically to: ece313.B@gmail.com.**
 - Explain all your work and include the code with comments.
 - Non-readable reports will be returned without a grade.
 - Write your names and group name on the report.
- **Tasks 0-1 and 2.1: due on Tuesday, October 8, 10:59 AM (before class).**
- **Task 2.2: due on Friday, October 11, 11:59 PM.**
- **Grading:**
 - **Task 1: 50%**
 - **Task 2: 50%**

MATLAB Tutorial

- Files:
 - Function and codes: *.m*
 - Files: *.mat*
- Import a data set into MATLAB workspace:
 - *load patient_data.mat*
- Export a data set from the MATLAB workspace:
 - *save result.mat*
 - *save result.mat X, Y, Z*
- Clear all the workspace environment:
 - *clear all*
- Clear the command window:
 - *clc*
- You can repeat the previous commands by arrow keys or from the command history window

MATLAB Tutorial (Cont.)

- Define variables in the command window:
 - $x = 1$
 - $y = X + Z$
 - $z = \cos(1)$
- To define arrays:
 - $T = [1, 2, 3, 4, 5]$
 - $T = 1:5$
 - $W = [1, 1.1, 1.2, 1.3, 1.4, 1.5]$
 - $W = 1:0.1:1.5$
 - *Labels = {'Heart Rate', 'Blood Pressure', 'Electrocardiogram'}*
 - $T = [1, 2, 3; 4, 5, 6]$ *(An array of 2 rows and 3 columns)*
- To see the values of variables, just type their name:
 - x
 - $W(1)$
 - $T(2:3)$

MATLAB Tutorial (Cont.)

- Define a vector of zeros or ones:
 - *zeros(1,10)* (A vector of 10 columns, all initialized to 0)
 - *ones(2,3)* (An array of 2 rows and 3 columns, all initialized to 1)
- Find the size of a vector or array:
 - *size(T)*
 - *size(W, 2)*
- Plotting a variable
 - *plot(x)*
 - *plot(t, x)*

MATLAB Resources

- MATLAB Tutorials and Learning Resources:
http://www.mathworks.com/academia/student_center/tutorials/launchpad.html
 - Watch short videos and look at MATLAB examples.
 - Read the user's guide.
- MATLAB help:
 - *help hist*
 - *doc hist*