Class Project

Binary Hypothesis Testing for Real-time Patient Monitoring

ECE 313 – Section G

Spring 2025

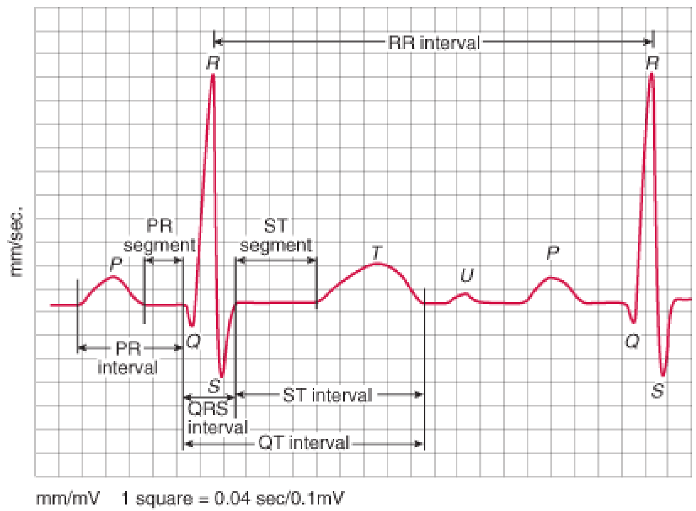
Task Objectives

- Hypothesis testing with single and pairs of features
  1. Learn how to use hypothesis testing to learn decision rules based on training data and use the generated rules to predict alarms on testing data
  2. Learn how to create ML and MAP decision rules by:
     - Creating the likelihood matrix based on training data
     - Creating the joint probability matrix from the likelihood matrix
     - Creating the likelihood matrix of joint observations from pairs of features
     - Generating the ML and MAP rules based on training data
  3. Learn how to evaluate decision rules using the conditional probabilities of false alarm and miss detection and probability of error
- Feature selection
  - Evaluate different criteria for selecting features, including:
    - Correlations between features
    - Correlations between features and golden alarms
    - Visualization of decision rules (ML and MAP)

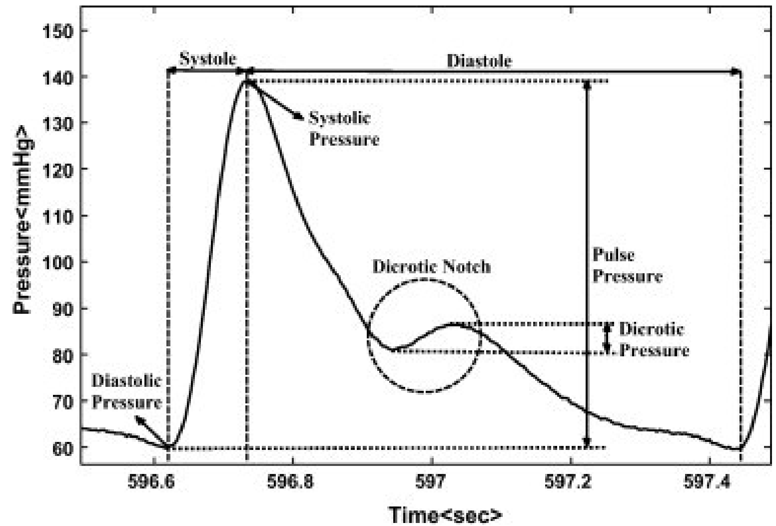
Class Project: New Data

- You are given a set of data collected from **9** different patients.
- Each data set includes **7** features calculated from the electrocardiogram (ECG) and blood pressure (ABP) signals collected from a given patient:

ECG Waveform



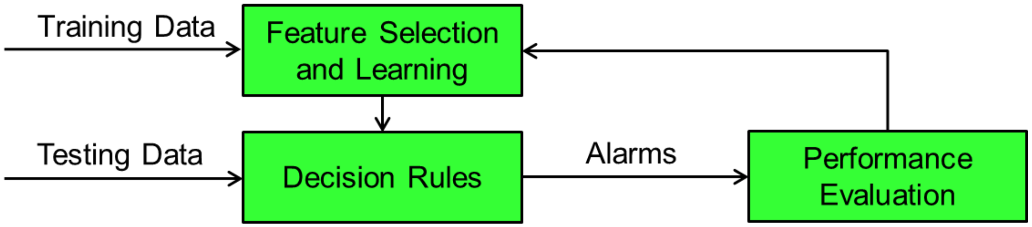
ABP Waveform



1. **Mean Area under the heartbeat**
2. **Mean R-to-R peak interval**
3. **Number of beats per minute (Heart Rate)**
4. **Peak to peak interval for Blood pressure**
5. **Systolic Blood Pressure**
6. **Diastolic Blood Pressure**
7. **Pulse Pressure**

Class Project: Task Summary

- Divide data into training and testing sets
- Learn ML and MAP decision rules based on training data:
  - Single features
  - Pairs of features
- Find the best pair of features for the least error in ML and MAP, using the training data for each patient and using at least two different criteria. Following are some example criteria. State your criteria precisely:
  - Correlations between features
  - Correlations between features and golden alarms
  - Visualization of decision rules (ML and MAP)
  - Weighted sum of pairs of features
- Use the ML and MAP rules using the features selected on testing data



Class Project: Task 3.0

(5 points) For each patient data provided do the following:

- Load each patient data file into Python, e.g using scipy.io.
- Each data file consists of two variables: *all_data* and *all_labels*:
- *all_data* is an array of 7 rows:
  - Each row corresponds to the following features, respectively:
    1. Mean Area under the heartbeat
    2. Mean R-to-R peak interval
    3. Number of beats per minute (Heart Rate)
    4. Peak to peak interval for Blood pressure
    5. Systolic Blood Pressure
    6. Diastolic Blood Pressure
    7. Pulse Pressure
  - Each column corresponds to a data sample.

    Note: Each patient data file has different number of samples.
- *all_labels* is a binary vector of golden alarms labeled by the physician
  - Each column corresponds to a data sample.

Class Project: Task 3.0

- Similar to Task 1.1, you need to first use ***numpy.floor(all_data)*** in order to convert data samples into *integers*. This will enable you in estimating the probability distributions as probability mass functions (pmf) by calculating the frequencies similarly to Task 1.1.

- Divide *all_data* and *all_labels* vectors to two sets of *training* and *testing* with training set being 2/3 of the data and testing set 1/3 of the data.

Hint: For example, if we have a dataset with 1500 entries:

```
train_data = all_data[:, :1000]
label_train = all_labels[:1000]
test_data = all_data[:, 1000:]
label_test = all_labels[1000:]
```

Class Project: Task 3.1

Let: ***H0*** the hypothesis that there is no patient abnormality (no alarm generated).

H1 the hypothesis that there is a patient abnormality (an alarm generated).

For each patient:

Use the *training* data set and its golden alarms (*label_training*), to:

a) (5 points) Calculate the prior probabilities of P(H1) and P(H0).

b) (5 points) Construct the likelihood matrices for each of the 7 features.

Hint 2: You can use the similar logic as in Task 1.1, but, remember that in this case you need conditioning and that ***probability values should be between 0 and 1***.

Hint 3: Remember that in the likelihood matrices, you need to have the same number of columns (possible values of the feature in the dataset) for both *H1* and *H0.* Find the range of the values that a variable takes by using *numpy.min* and *numpy.max*.

Final Project: Task 3.1 (Cont'd)

c) (5 points) Show your results by generating a separate figure for each patient, consisting of 7 *subplots* corresponding to the 7 features. In each subplot of each figure, plot the conditional pmf under each of the hypotheses *H0* and *H1*. Use *matplotlib.pyplot.legend* in Python to distinguish between the two pmf's in the subplot.

d) (5 points) Calculate the ML and MAP decision rule vectors.

Hint 4: To be definite, break ties in favor of *H1*. For example:

If *H1_pmf(i) ≥ H0_pmf(i)* then *ML_vector(i) = 1*.

e) (5 points) **Save the results of Task 3.1** in the form of a 9-by-7 cell array, **called HT_table_array**, with each cell representing a two-dimensional array in the following format:

Hint: *HT_table_array = [[None for _ in range(7)] for _ in range(9)]*

HT_table_array[patient1][feature1] = HT_table[feature1]

The range of values that feature X takes.

X = i	P(X=i | H1)	P(X=i | H0)	ML Predicted Label	MAP Predicted Label

Final Project: Task 3.2

a) (5 points) Use the *HT_table_array* calculated in Task 3.1 part e, to generate alarms based on each of the ML and MAP decision rules for the *testing* data set.

b) (5 points) Use *label_testing* golden alarms to evaluate each of the ML and MAP decision rules, by calculating:

1. The conditional probability of false alarm
2. The conditional probability of missed detection
3. The probability of error

Hint: Remember:

P(False Alarm) = P(Decision rule declares an alarm | Physician indicates no abnormality)

P(Miss Detection) = P(Decision rule declares no alarm | Physician indicates an abnormality)

P(Error) = P(Decision rule declares an alarm AND Physician indicates no abnormality)+
P(Decision rule declares no alarm AND Physician indicates an abnormality)

Save the results of Task 3.2 in the form of a 9-by-7 cell array, **called Error_table_array**, with each cell representing a 2-by-3 array in the following format:

	P(False Alarm)	P(Miss Detection)	P(Error)
ML Rule			
MAP Rule			

Final Project: Task 3.2

In the rest of the project you will perform ML and MAP hypothesis testing using a **pair of features**.

Use the *training* data set and its golden alarms (*label_training*), to:

- Find the **best pair of features** that would achieve the lowest probability of error for each of the ML and MAP rules for 3 patients that you think are best. i.e., remember you will be choosing one pair per patient

- Justify how each of your criteria relates to the expected performance in the hypothesis testing by showing the plots and results from your analysis above.

- Choose a set of possible pairs of features to be used for hypothesis testing based on pairs of features in the next task.

Final Project: Task 3.2 (Cont'd)

- (10 points) Use **at least two** of the following techniques or other techniques to find the best pair of features to be used with ML and MAP decision rules as explained below:

  1. Find the top two features with the lowest ML and/or MAP errors from Task 3.1,3.2.

  Hint: For example if feature f1 had the lowest ML error you may use it as one of the features in the pair used for the ML rule. If f2 had the lowest MAP error, you may use it in the pair for MAP.

  2. Analyze the correlation between each feature and golden alarms from Task 3.2.

  Hint: Plot the features and ML and MAP alarms generated based on each feature in Task 3.2, along with *label_testing* golden alarms provided. Plot in one figure to visualize the correlation.

  3. Analyze the correlation between different pairs of features.

  Hint: If two features are highly correlated, having them in a pair won't provide more information for decision making. To find correlation between features f1 and f2:

  e.g. *np.corrcoef(train_data[f1], train_data[f2])[0, 1] gives you correlation*

  4. Use the weighted sum of pairs of the feature values to generate new candidate features