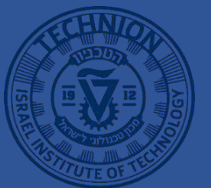


#L03-Linear models for regression

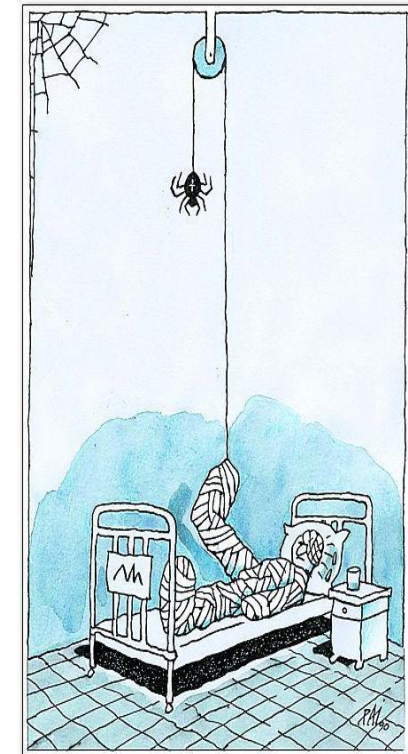
Technion-IIT, Haifa, Israel

Asst. Prof. Joachim Behar
Biomedical Engineering Faculty, Technion-IIT
Artificial intelligence in medicine laboratory (AIMLab.)
<https://aim-lab.github.io/>
Twitter: @lab_aim



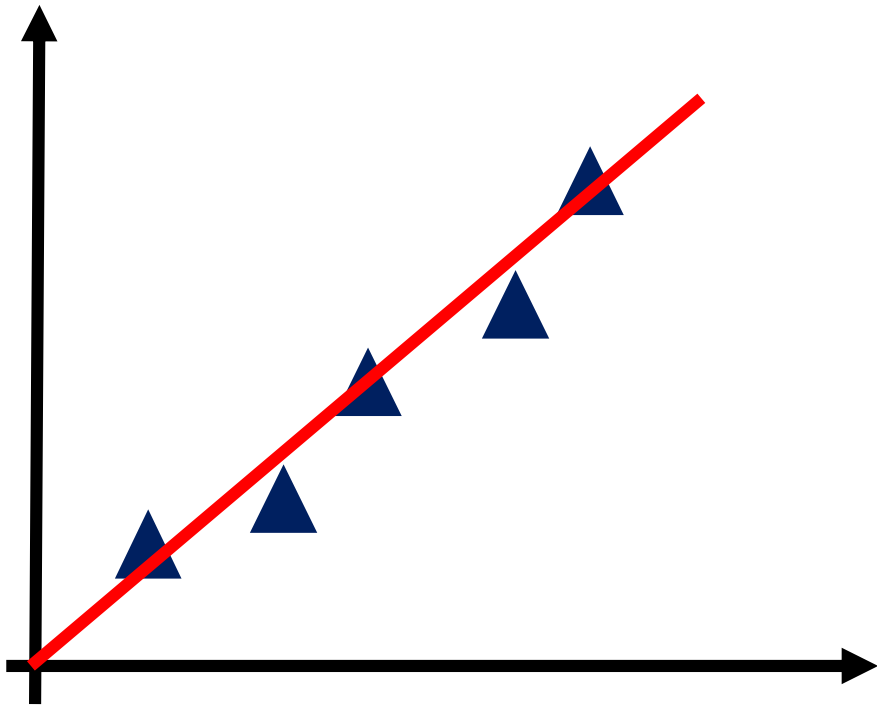
The Problem

- Given the information x , we want to predict event y
- E.g. Our patient's respiration rate is 14, heart rate is 72, GCS is 15.
 - *How long will this patient stay in the hospital?*



The Problem

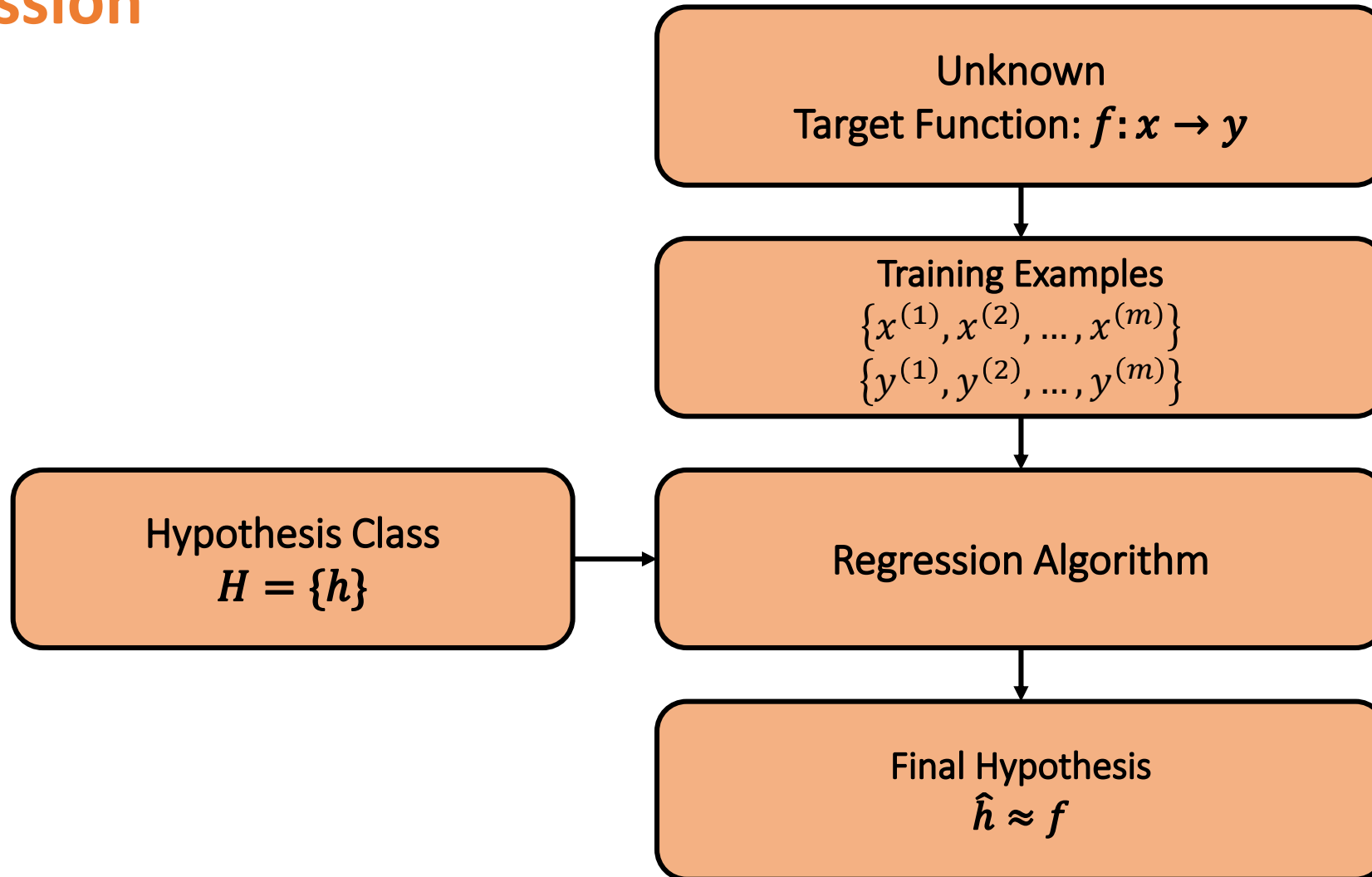
Regression



Estimate relationships among usually continuous variables.



Regression



Defining the Problem

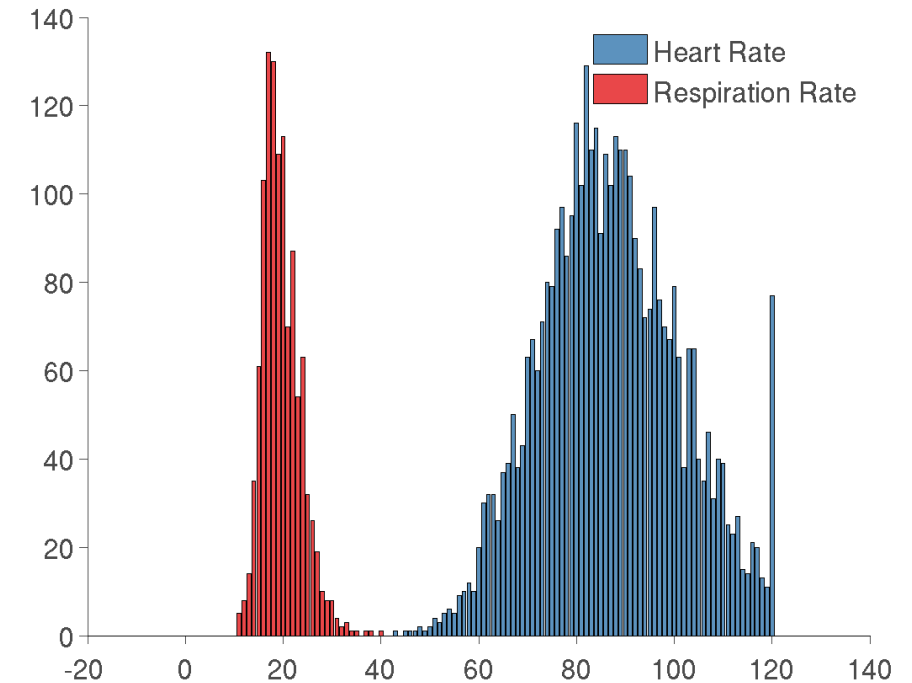
- Features, targets, function:

- Features: $X = \begin{bmatrix} HR_1^{(1)} & RR_{n_x}^{(1)} \\ \dots & \dots \\ HR_1^{(m)} & RR_{n_x}^{(m)} \end{bmatrix} \in \mathbb{R}^{m \times n_x}$

- Target: $\underline{y} = [y^{(1)}, \dots, y^{(n)}]$

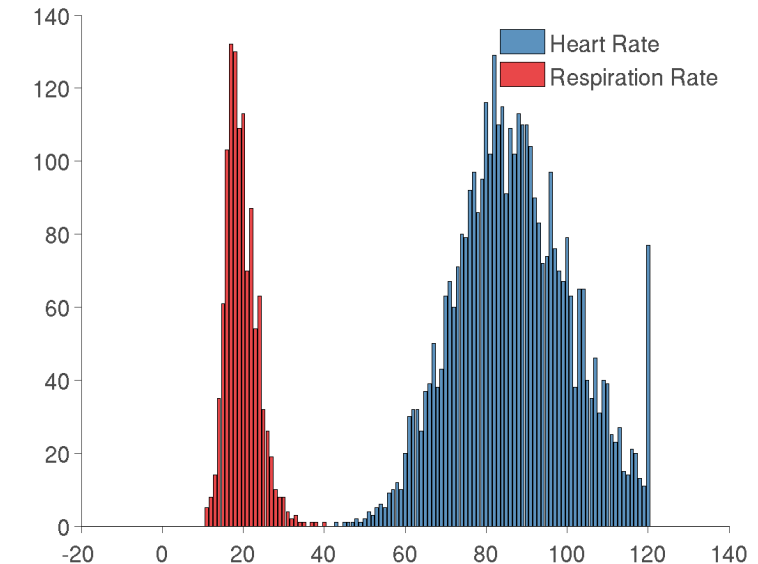
- Function: $\underline{y} = f(X)$

- Here the hypothesis functions is assumed **linear**.



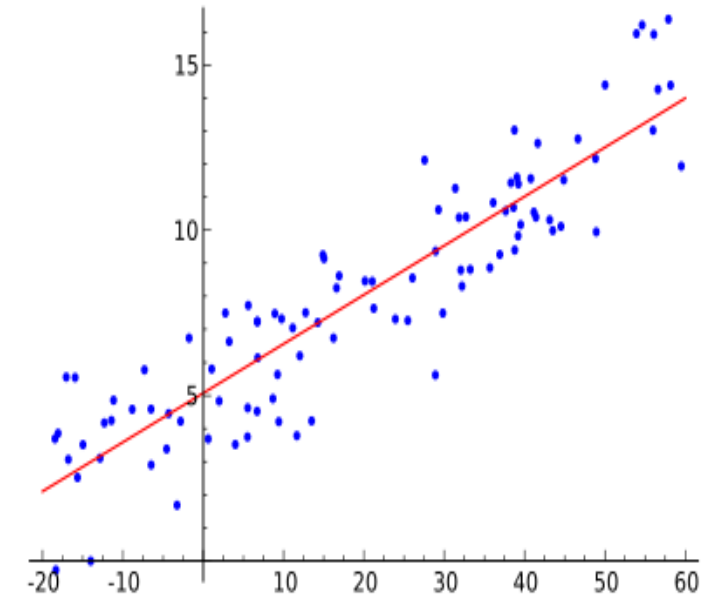
Defining the Problem

- On the example:
 - We can imagine that say, higher heart rates (HR) are bad.
 - We can imagine that lower respiration rates (RR) are bad.
 - $y = w_1 HR + w_2 RR$
 - How do you imagine w_1 and w_2 to look like?



Linear Regression - Intuition

- More generally:
 - $\underline{y} = f(X) = Xw$
 - $w \in \mathbb{R}^{n_x}$
 - $X \in \mathbb{R}^{m \times n_x}$
- Linear regression aims to predict *an independent variable* from one or more *dependent variables*.



Linear Regression - Intuition

- Formally, **linear regression** involves solving for w :
 - $\underline{y} = w_0 + w_1x_1 + w_2x_2 + \dots + w_nx_n$
- We need a set of m examples for that purpose:
 - $x^{(i)} = [x_1, \dots, x_{n_x}] \rightarrow y^{(i)}$
 - This is our **training set**!

Linear Regression - Intuition

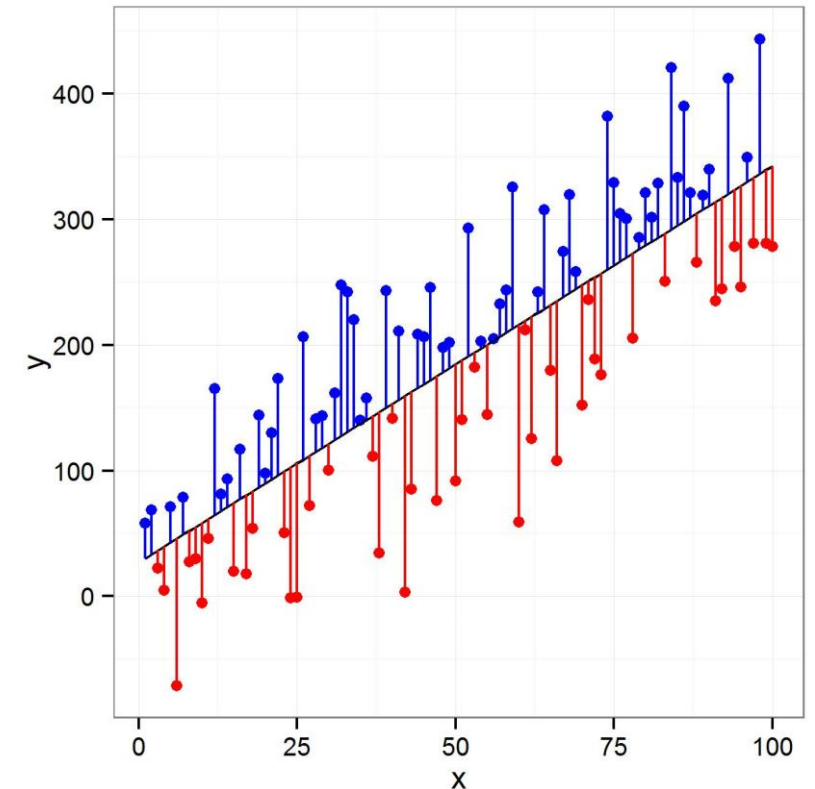
- How do we estimate the coefficients?
- Intuitively, we can consider an example:
 - $\begin{bmatrix} 3 \\ 7 \end{bmatrix} = \begin{bmatrix} 3 & 8 \\ -1 & 4 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$
- Simple enough with matrix math: invert X and solve:
 - $\begin{bmatrix} w_1 \\ w_2 \end{bmatrix} = \begin{bmatrix} 3 & 8 \\ -1 & 4 \end{bmatrix}^{-1} \begin{bmatrix} 3 \\ 7 \end{bmatrix}$
- But it's not always so simple!
 - $\begin{bmatrix} 1 \\ 3 \end{bmatrix} = \begin{bmatrix} 2 & 3 \\ 6 & 9 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}.$

Linear Regression - Intuition

- How do we estimate the coefficients without directly inverting X ?
- The best we can do is minimize some kind of error.
- For example we like **the mean square error (MSE)**!

- $\min_w \left(\left\| \underline{y} - Xw \right\|_2^2 \right)$

- $J(w) = \frac{1}{m} \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$



Mathematical proof

Linear Regression – Normal equation

- In the situation where the matrix is not square, we can't directly take the inverse:
 - $\underline{y} = Xw$
- But because we're clever, we noted that $X^T X$ is a **positive semi definite** matrix and that these matrices are invertible!
 - $X^T X w = X^T \underline{y}$
 - $w = (X^T X)^{-1} X^T \underline{y}$
 - $(X^T X)^{-1} X^T$: pseudo inverse of X
- Turns out minimizing the mean square error is equivalent solving the “**Normal equation**”* ➔ We will demonstrate that now.

*For bonus points, refer to this as the Moore-Penrose pseudoinverse. Name dropping dead mathematicians makes you sound smart. ¹²

Linear Regression – Normal equation

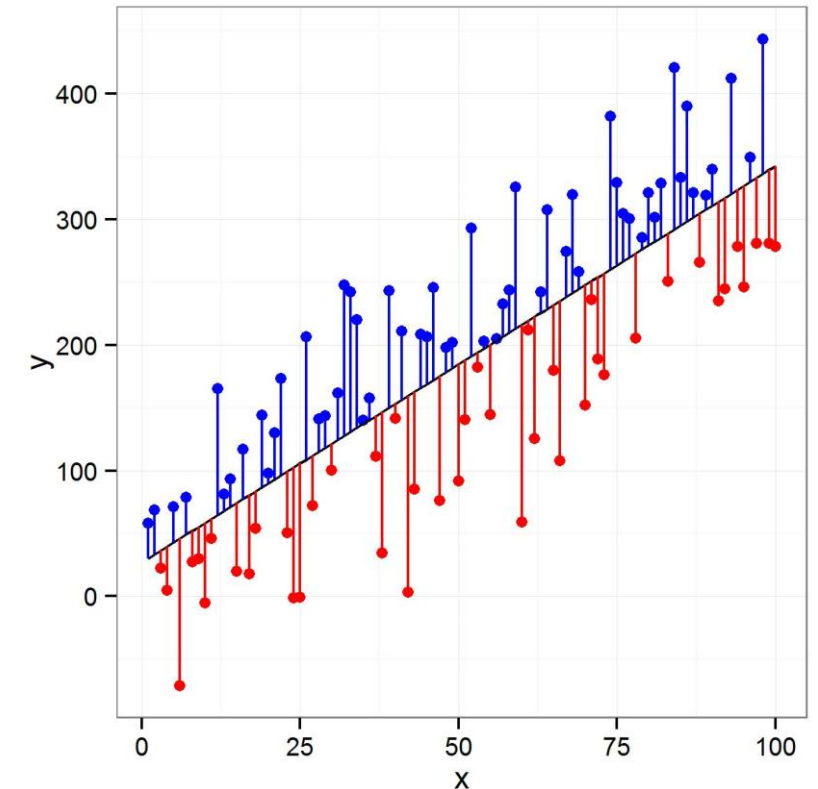
- To recap:
 - Hypothesis function in linear regression:
 - $y = w_0 + w_1x_1 + \dots + w_nx_n$
 - Has a solution that looks like:
 - $w = (X^T X)^{-1}X^T \underline{y}$ (Normal equation).
 - We now want to prove that the **Normal equation** is a solution to the **linear regression** problem with a **mean square cost function**: $\min_w \left(\left\| \underline{y} - Xw \right\|_2^2 \right)$
- That is, we want to minimize the least square cost function:
 - $J(w) = \frac{1}{m} \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$

Mathematical Proof: Using Matrix Calculus

- We want to minimize the least square cost function:
 - $J(w) = \frac{1}{m} \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$
 - $J(w) = (\underline{X} \underline{w} - \underline{y})^T (\underline{X} \underline{w} - \underline{y})$

$$= (\underline{X} \underline{w})^T \underline{X} \underline{w} - (\underline{X} \underline{w})^T \underline{y} - \underline{y}^T (\underline{X} \underline{w}) + \underline{y}^T \underline{y}$$

$$= (\underline{w}^T \underline{X}^T \underline{X} \underline{w} - 2(\underline{X} \underline{w})^T \underline{y} + \underline{y}^T \underline{y})$$
 - $\frac{\partial J}{\partial \underline{w}} = 2\underline{X}^T \underline{X} \underline{w} - 2\underline{X}^T \underline{y} = 0$
 - $\underline{X}^T \underline{X} \underline{w} = \underline{X}^T \underline{y} \Leftrightarrow \underline{w} = (\underline{X}^T \underline{X})^{-1} \underline{X}^T \underline{y}$
- We have shown that minimizing the mean square cost function in linear regression is equivalent to solving the normal equation.



Mathematical Proof: Probabilistic Derivation

- We now want to make this mathematical proof by taking a probabilistic approach.
- Probability: measure of the **likelihood** of an event to happen.
- Target scalar y is given by a deterministic function $f(x, w)$ with an additive zero mean Gaussian noise:
 - $y = f(x, w) + \epsilon$
 - $\epsilon \sim \mathcal{N}(0, \beta^{-1}), \beta = 1/\sigma^2$
 - Thus is a zero mean Gaussian random variable.
 - For one example: $p(y|x, w, \beta) = \mathcal{N}(y|f(x, w), \beta^{-1}) = \mathcal{N}(y|w^T x, \beta^{-1})$
- We can write the likelihood function:
 - $p(\underline{y}|X, w, \beta) = \prod_{i=1}^m \mathcal{N}(y^{(i)}|w^T x^{(i)}, \beta^{-1}).$

Mathematical Proof: Probabilistic Derivation

- In supervised learning problems such as regression (and classification) we are not seeking to model the distribution of the input variables. Thus X will always appear in the set of conditional variables and so we can simplify the notation:

- $p(\underline{y}|X, w, \beta) \rightarrow p(\underline{y}|w, \beta)$

- Taking the logarithm of the likelihood function:

- $$\ln\left(p(\underline{y}|w, \beta)\right) = \sum_{i=1}^m \ln(\mathcal{N}(y^{(i)}|w^T x^{(i)}, \beta^{-1}))$$
$$= \frac{m}{2} \ln(\beta) - \frac{m}{2} \ln(2\pi) - \beta \frac{1}{2} \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$$

Sum-of-squares error function



Mathematical Proof: Probabilistic Derivation

- $$\ln \left(p \left(\underline{y} | w, \beta \right) \right) = \frac{m}{2} \ln(\beta) - \frac{m}{2} \ln(2\pi) - \beta \frac{1}{2} \sum_{i=1}^m (y^{(i)} - w^T x^{(i)})^2$$

Mathematical Proof: Probabilistic Derivation

- We now take the derivative of the log likelihood:
 - $\nabla \ln(p(y|w, \beta)) = \beta \sum_{i=1}^m (y^{(i)} - w^T) x^{(i)T}$
- Setting the gradient to zero leads to:
 - $\sum_{i=1}^m (y^{(i)} - w^T x^{(i)}) x^{(i)T} = 0$
 - $\sum_{i=1}^m y^{(i)} x^{(i)T} - w^T \sum_{i=1}^m x^{(i)} x^{(i)T} = 0$
 - $w = (X^T X)^{-1} X^T \underline{y}$

Mathematical Proof: Probabilistic Derivation

- We assumed that $\epsilon \sim \mathcal{N}(0, \beta^{-1})$
- In this particular case we showed that the maximum likelihood estimate is equal to the ordinary least square (OLS) estimate.
- We could also estimate β and through that learn what is the noise representation.

Sequential Learning

Sequential Learning

- The **normal equation** is a closed form solution. It involves processing the entire training set in one go. This can be computationally costly for large datasets.
- **Sequential algorithm**/on-line algorithm can be useful: data points are considered one at a time and model parameters are updated after each such presentation.
- It is also useful when observations are arriving in a continuous stream and prediction must be made before all prediction are seen.

Sequential Learning

- Stochastic gradient descent:
 - $w(n + 1) = w(n) + \eta \cdot \nabla J(n)$
- Assuming a sum of squared error:
 - $w(n + 1) = w(n) + \eta \cdot (y(n) - w^T(n) \cdot x(n))x(n)$
 - This is known as the **least mean square (LMS) algorithm**.
 - η is the learning rate - must be chosen carefully to ensure convergence.
 - w is initialized to some starting values
 - $J(n)$ the error of the n^{th} incoming data point.

Sequential Learning - Example

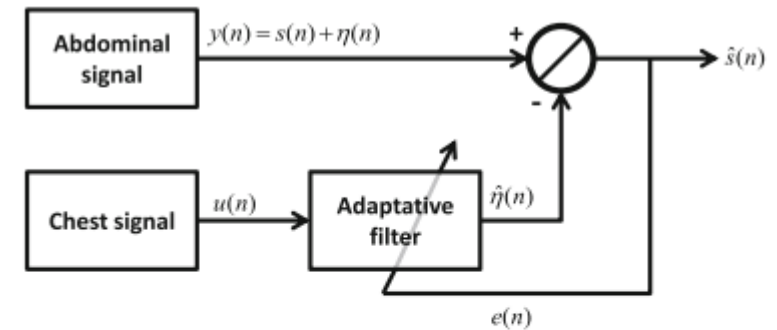
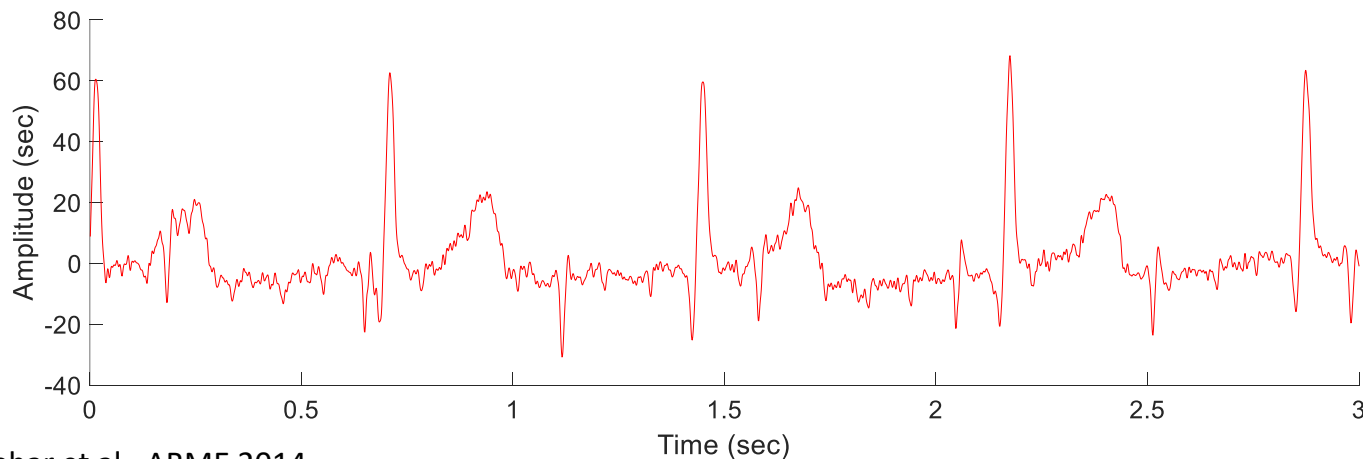
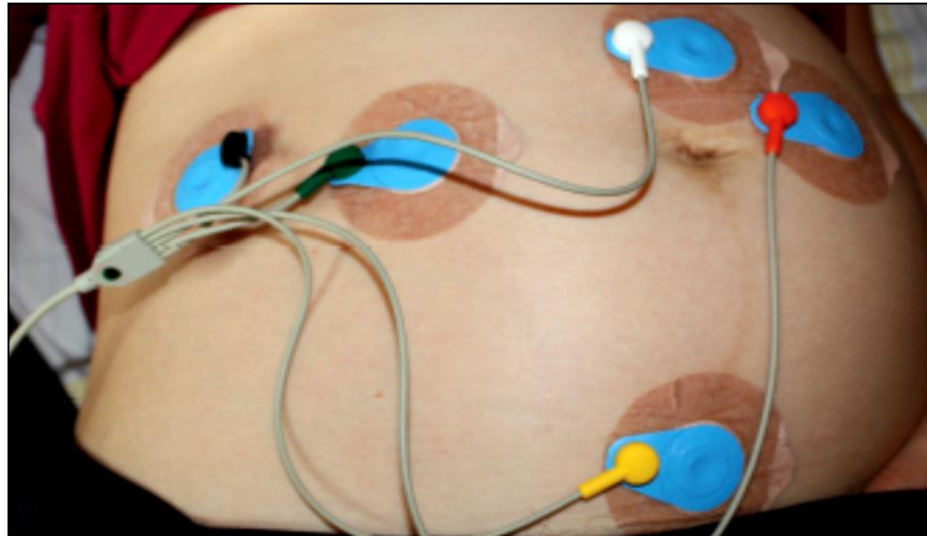


FIGURE 2. Adaptive noise canceling block diagram in the case of one reference input $u(n)$. On the diagram : the FECG $s(n)$, the noise $\eta(n)$, the abdominal ECG $y(n) = s(n) + \eta(n)$, the chest signal $u(n)$, the estimated noise $\hat{\eta}(n)$, the estimation error $e(n)$ and the output signal $\hat{s}(n)$. n corresponds to a time index.

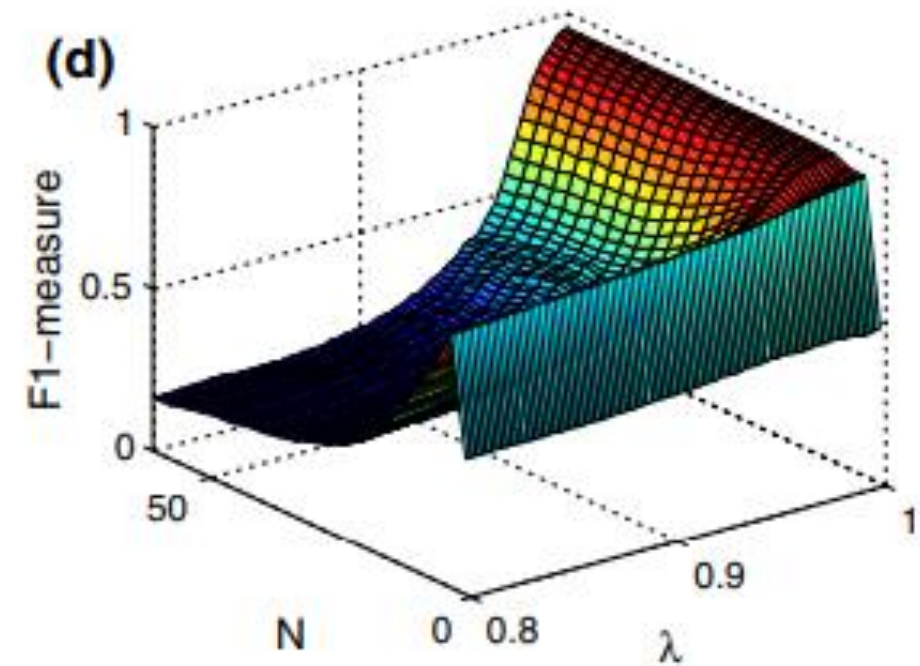
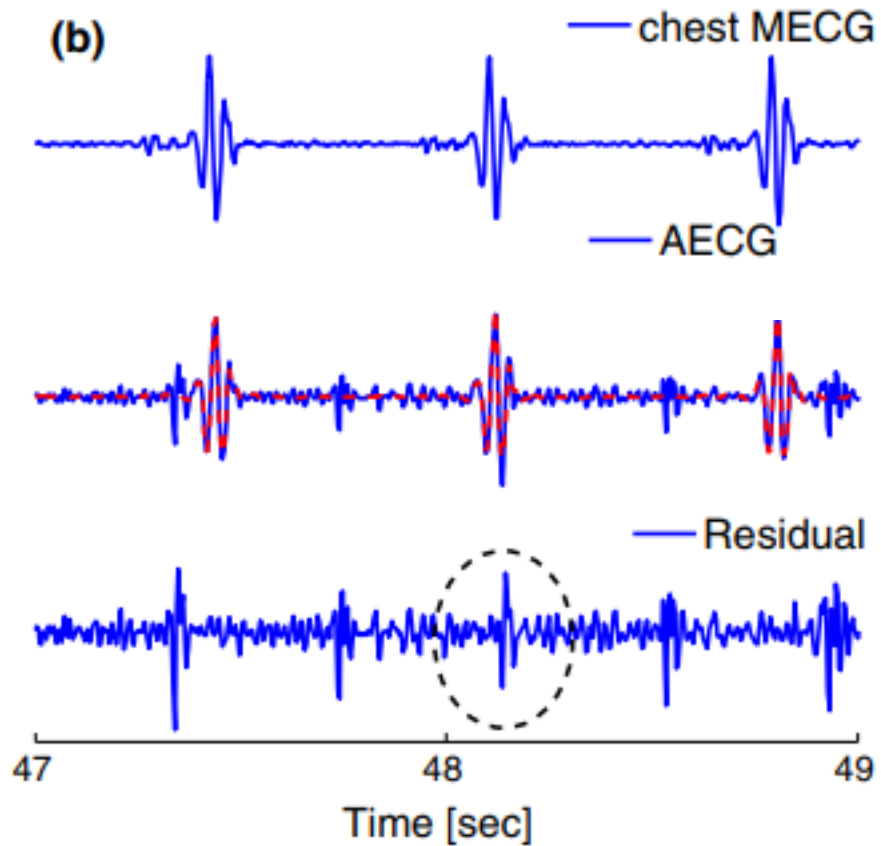
$$\hat{\eta}(n) = \underline{\mathbf{w}}^T(n-1)\underline{\mathbf{u}}(n) \quad (1)$$

$$e(n) = y(n) - \hat{\eta}(n) \quad (2)$$

$$\underline{\mathbf{w}}(n) = \underline{\mathbf{w}}(n-1) + \mu e(n)\underline{\mathbf{u}}(n) \quad (3)$$

Sequential Learning - Example

- Dataset 1 for training and dataset 2 for test.



Take Home

- **Linear regression:** predict an independent variable from one or more dependent variables.
- **Algebraic and Probabilistic derivation** of the solution.
 - Differences in assumptions and approach to the same problem.
 - **Normal equation** - closed form solution.
 - Probabilistic derivation - also provides an estimate of the noise distribution.
- **Stochastic gradient descent** - sequential learning.

Next Lecture

- Regression versus Classification?
- The approach you take depends on the question you ask:
 - How long will this patient stay in ICU? → **regression**.
 - Will this patient survive from his ICU stay? → **classification**.
- In the next lecture we will talk about **Linear Models for Classification**.
- Important note: defining your question i.e. whether you are dealing with a regression or classification problem is critical before starting any development!

References

- [1] Oxford, CDT course 2015
- [2] Pattern recognition and Machine Learning. Christopher M. Bishop. 2006 Springer Science.
- [3] <https://eli.thegreenplace.net/2014/derivation-of-the-normal-equation-for-linear-regression/>