

Statistical Analysis of Peripheral Venous Pressure Signals for Estimating Body Fluid Status

Ph.D. Candidacy Exam

Md Abul Hayat

Electrical Engineering
University of Arkansas
Fayetteville, AR 72701, USA

Committee Members

Jingxian Wu, Ph.D. (*Chair*)

Morten O. Jensen, Ph.D.

Roy A. McCann, Ph.D.

Yue Zhao, Ph.D.

Outline

- 1 Motivation
- 2 Venous Physiology Predicts Dehydration in the Pediatric Population
- 3 Unsupervised Anomaly Detection in PVP Signals
- 4 Future Works
- 5 Additional Contributions

Motivation

- Dehydration or loss of intravascular blood volume is a common and potentially life-threatening condition
- Dehydration affects 30 million children annually and accounts for 400,000 pediatric emergency room visits in the United States.
- Up to 10% of all US hospital admissions of children <5 years of age are because of diarrhea and dehydration [WMB04]
- Assessment of fluid volume status remains an elusive problem in clinical medicine
- There is no standardized measurement for intravascular volume in adults or children
- This necessitates the development of technologies that would accurately assess the volume status of a patient to guide resuscitation and treatment

Peripheral Venous Pressure Signal

- **P**eripheral **V**enous **P**ressure (PVP) Signal
 - Collected from peripheral veins
- The signal collection method is minimally invasive
- It can be easily collected using ubiquitously regular catheters
- The PVP signal is a good representative of blood-circulation (cardiovascular) system
- PVP is strongly correlated with other vital invasive signals like Central venous pressure (CVP) and Jugular vein pressure (JVP) [HEY+06]

Data Acquisition System

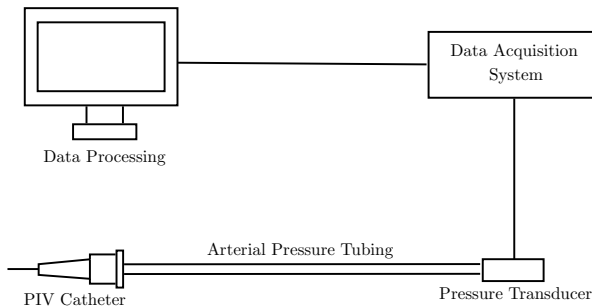


Figure 1: Schematic diagram of the data acquisition system. Peripheral intravenous (PIV) catheter is inserted into peripheral vein of a patient. Source: [HWB+20]

Dataset Description

- Total 18 patients of hypertrophic pyloric stenosis
- Electrolytes are used as a marker of resuscitation
 - Hypovolemic/Dehydrated
 - $\text{Cl}^- < 100 \text{ mmol/L}$
 - $\text{HCO}_3^- \geq 30 \text{ mmol/L}$
 - Resuscitated/Euvolemic/Hydrated
 - $\text{Cl}^- \geq 100 \text{ mmol/L}$
 - $\text{HCO}_3^- < 30 \text{ mmol/L}$

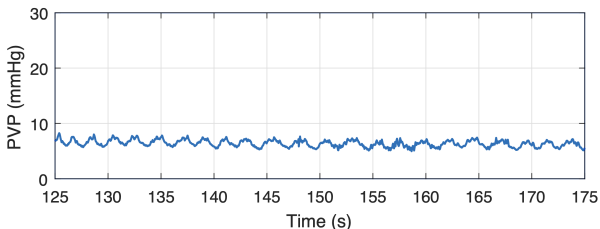


Figure 2: PVP signal

Data Pre-processing

- Splitting each patient's data into non-overlapping *windows* of 10s
- Hypovolemic windows: 329
- Resuscitated windows: 343
- FFT was performed over the time domain signals in each window
- Frequency domain resolution of $1/10 = 0.1$ Hz
- Each window contains 200 frequency domain samples between 0 and 19.9 Hz (inclusive)
- No filtering used
- Anomalous segments were removed manually

Logistic Regression with Regularization

- Logistic regression is a binary classifier
- If i -th frequency domain window is X_i then

$$X_i = [1, f_0, f_{0.1}, \dots, f_{19.9}]$$

- And corresponding label Y_i is defined as

$$Y_i = \begin{cases} 1, & \text{Hypovolemic/Dehydrated} \\ 0, & \text{Resuscitated/Euvolemic/Hydrated} \end{cases}$$

- This is a supervised classification problem

Logistic Regression with Regularization

- The aim is to design β during training such that

$$P(Y_i = 1) = \frac{1}{1 + e^{-\beta^T X_i}}$$

$$P(Y_i = 0) = \frac{e^{-\beta^T X_i}}{1 + e^{-\beta^T X_i}}$$

- Here, β is a 201-dimension regression coefficient vector
- Once β is trained and validated, it can be used on testing dataset

Logistic Regression with Regularization

- β is estimated by minimizing the following loss function $\ell(\beta)$

$$\ell(\beta) = \sum_{i=1}^n [(1 - y_i)\beta^T X_i + \log(1 + e^{-\beta^T X_i})] + \lambda P_\alpha(\beta)$$

$$P_\alpha(\beta) = \frac{1 - \alpha}{2} \|\beta\|_2^2 + \alpha \|\beta\|_1$$

- Here, n is the total number of training windows
- $\|\beta\|_1$ and $\|\beta\|_2$ are the L_1 and L_2 norms of β respectively
- Two special cases of Elastic-net $\alpha \in [0, 1]$ are

$$\alpha = \begin{cases} 1, & \text{LASSO} \\ 0, & \text{Ridge} \end{cases}$$

- LASSO is **L**east **A**bsolute **S**hrinkage & **S**election **O**perator

Logistic Regression with Regularization

- 70% windows from each category were chosen as the training and validation data
- 232 windows with $Y = 1$ and 242 windows with $Y = 0$ were used for training and validation
- The remaining 30% were used for testing
- To tune λ , 5-fold cross-validation (CV) is applied training data
- Within each CV, 80% of the training data have been used for training purpose and 20% (validation) have been used to find the deviance for selecting proper λ
- $\lambda = 0.0055$ gives the minimum deviance during CV with LASSO
- The optimum λ forced 158 of the 201 coefficients of β to be zero

Classification Results

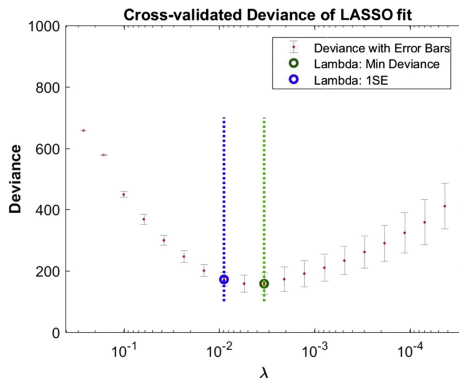


Figure 3: Cross-validation for optimizing the tuning parameter λ . The minimum deviance is obtained at $\lambda = 0.0055$.

Classification Results

- Performance metrics

- Sensitivity/TPR = $\frac{TP}{TP+FN}$

- Specificity/TNR = $\frac{TN}{TN+FP}$

- Window level classification

α	Training Sensitivity(%)	Training Specificity(%)	Testing Sensitivity(%)	Testing Specificity(%)	Non-zero Coefficients
0.0001	94.40	95.87	97.94	93.07	201
0.5	99.57	99.59	96.91	93.07	73
0.75	99.57	100	96.91	92.08	64
1 (LASSO)	99.57	99.17	97.95	93.07	43

- Patient level classification

- Majority voting on aggregate window decisions
 - Sensitivity(%) = 100%, Specificity(%) = 100%

Kolmogorov–Smirnov Two-sample Test

- Kolmogorov–Smirnov (KS) two-sample test is a non-parametric test
- It is used to identify if two different samples are coming from the same distribution or not
- In frequency domain data, two-sample test has been performed over the frequency range from 1.5 Hz to 4.5 Hz.
- H_0 : Samples follow same distribution at a frequency f Hz
 H_1 : Samples follow different distribution at f Hz

Kolmogorov–Smirnov Two-sample Test

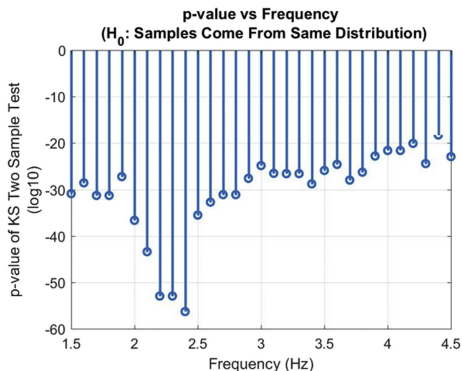


Figure 4: p -value of the Kolmogorov-Smirnov two-sample test over frequency range 1.5 to 4.5 Hz.

Kolmogorov–Smirnov Two-sample Test

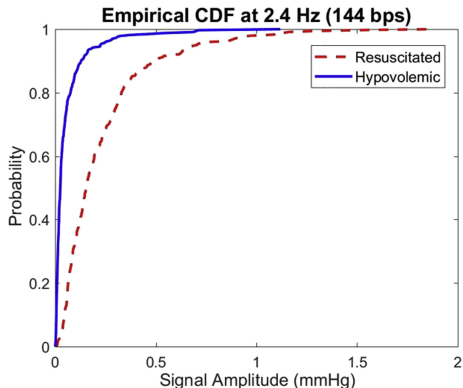


Figure 5: The empirical cumulative distribution functions (CDFs) of data from hypovolemic or resuscitated patients at 2.4 Hz.

Amplitude: Hydrated

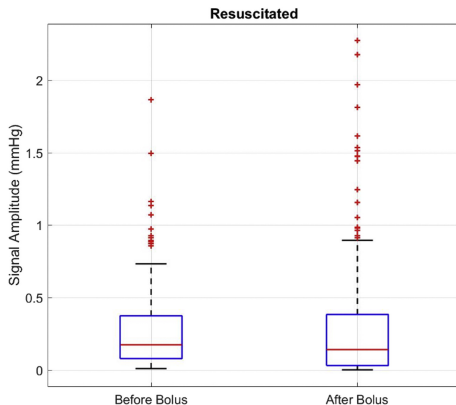


Figure 6: Box plot of peripheral venous pressure (PVP) signal amplitude at 2.4 Hz for resuscitated patients before and after bolus. The signal amplitude does not change significantly. The outliers are plotted using “+” symbol in the figure.

Amplitude: Dehydrated

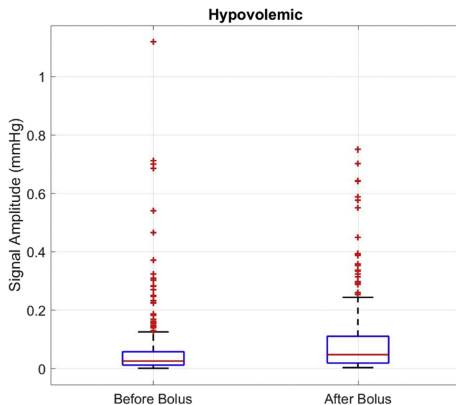


Figure 7: Box plot of peripheral venous pressure (PVP) signal amplitude at 2.4 Hz for hypovolemic patients before and after bolus. The signal amplitude changes significantly due to bolus. The outliers are plotted using “+” symbol in the figure.

Artery-Vein Crosstalk

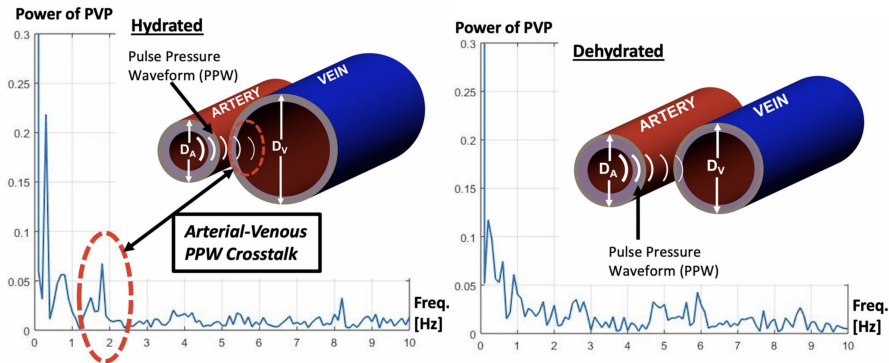


Figure 8: The power spectral density (PSD) of peripheral venous pressure (PVP) for the hydrated patient (left) identifies a peak at frequencies around the heart rate (red dotted line). In this example, the peak is shown at approximately 1.8 Hz = 108 bpm. In the same patient during dehydration (right), this phenomenon does not exist.

Unsupervised Anomaly Detection in PVP Signals

- PVP signals are highly susceptible to motion and noise artifacts
- To detect anomalies, we propose a two step model
- **Step-1:** PVP signals are represented and modeled by using dynamic linear models (DLM)
- **Step-2:** The DLM-based Kalman filter prediction residuals are modeled by using a hidden Markov model (HMM)
- DLMs are special case of state space models being Gaussian, linear and continuous

PVP Signal Corruption

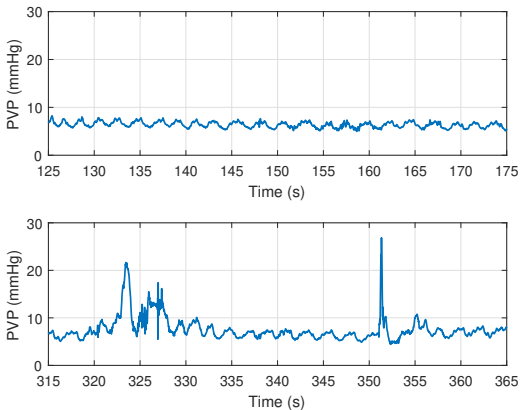


Figure 9: Exemplary PVP signal (a) without anomaly and (b) with anomaly.
Image: [HWB+20]

DLM Model

- Define, PVP signal samples y_t , $\forall t = 1, 2, \dots, T$
- To simplify notation, $\mathbf{y}_{1:T} = [y_1, y_2, \dots, y_T]$
- We assume, y_t is associated with a time-varying (latent) state, θ_t by

$$y_t = F\theta_t + v_t;$$

$$\theta_t = G\theta_{t-1} + w_t.$$

- Here, F and G are constants for a given patient
- $v_t \sim \mathcal{N}(0, \sigma_v^2)$, $w_t \sim \mathcal{N}(0, \sigma_w^2)$ and $v_t \perp w_t$.

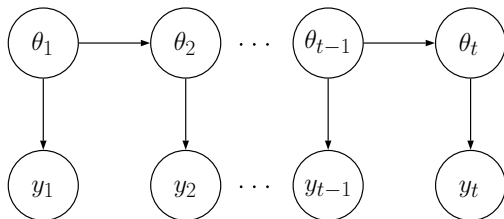


Figure 10: Dependence structure of dynamic linear model. Here, θ_i 's are forming a first-order Markov chain. Also, θ_{i+1} and y_i follow a Gaussian distribution depending on θ_i under a linear relationship. $\{\theta_i\}$ and $\{y_i\}$ are continuous random variables. Image: [HWB+20]

Kalman Filter

- Given all previous observations $\mathbf{y}_{1:t-1}$, Kalman filter is used to estimate y_t and the state variable θ_t
- $\hat{\theta}_{t|\tau}$ as the estimation of θ_t by using $\mathbf{y}_{1:\tau}$ with $\tau = t - 1$ or t
- $\hat{y}_{t|\tau}$ as the estimation of y_t by using $\mathbf{y}_{1:\tau}$ with $\tau = t - 1$
- We define: $Q_{t|\tau} = \mathbb{E}[|\hat{y}_{t|\tau} - y_t|^2]$ and $R_{t|\tau} = \mathbb{E}[|\hat{\theta}_{t|\tau} - \theta_{t|\tau}|^2]$
- The values of $\hat{\theta}_{t|\tau}$, $R_{t|\tau}$, $\hat{y}_{t|\tau}$, and $Q_{t|\tau}$ can be iteratively updated by using the Kalman filter, and details are in the following slide

DLM Model Parameters

- The DLM is represented by parameter set, $\{F, G, \sigma_v^2, \sigma_w^2, \hat{\theta}_{0|0}, R_{0|0}\}$
- The parameters are estimated by using the maximum likelihood estimation algorithm L-BFGS-B [PA10]
- Once the parameters are estimated, Kalman filter can then be applied to track and estimate the dynamic evolution of the data in the time

Kalman Filter Algorithm

Algorithm 1 Kalman Filter

- 1: **Input:** Discrete time dataset $\mathbf{y}_{1:T}$;
- 2: Estimate $\{F, G, \sigma_v^2, \sigma_w^2, \hat{\theta}_{0|0}, R_{0|0}\}$ using L-BFGS-B algorithm on $\mathbf{y}_{1:T}$; initialize $t = 0$.
- 3: **do**
- 4: $t \leftarrow t + 1$;
- 5: Prediction of θ_t :
$$\hat{\theta}_{t|t-1} = G\hat{\theta}_{t-1|t-1}$$
$$R_{t|t-1} = G^2 R_{t-1|t-1} + \sigma_w^2$$
- 6: Prediction of y_t :
$$\hat{y}_{t|t-1} = F\hat{\theta}_{t|t-1}$$
$$Q_{t|t-1} = F^2 R_{t|t-1} + \sigma_v^2$$
- 7: Calculate prediction residual: $x_t = y_t - \hat{y}_{t|t-1}$;
- 8: Update estimation of θ_t :
$$\hat{\theta}_{t|t} = \hat{\theta}_{t|t-1} + FR_{t|t-1}Q_{t|t-1}^{-1}x_t$$
$$R_{t|t} = R_{t|t-1} - F^2R_{t|t-1}^2Q_{t|t-1}^{-1}$$
- 9: **while** $t \leq T$
- 10: **Output:** Prediction residual $\mathbf{x}_{1:T}$.

Modeling PVP Prediction Residuals with HMM

- Residual x_t is associated with a binary hidden state $s_t \in \{0, 1\}$, where
 - $s_t = 0$ indicates normal data, and $s_t = 1$ indicates anomalies.

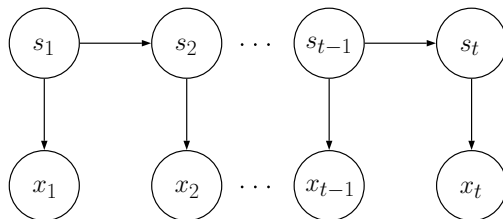


Figure 11: Dependence structure of first-order hidden Markov model. Unlike fig. 10, here $\{s_i\}$ are discrete random variables and x_i follows a Gaussian distribution depending on s_i . Image: [HWB+20]

HMM Formulation

- $\mathbf{x}_{1:T} = [x_1, x_2, \dots, x_T]$
- $a_{ij} = \Pr(s_{t+1} = j | s_t = i)$, for $i, j \in \{0, 1\}$ and $\mathbf{A} = [a_{ij}]$
- $\pi_k = \Pr(s_1 = k)$ for $k \in \{0, 1\}$ and $\boldsymbol{\pi} = [\pi_0, \pi_1]^T$
- $x_t | (s_t = k) \sim \mathcal{N}(\mu_k, \sigma_k^2)$, for $k \in \{0, 1\}$ and $\mathcal{B} = \{\mu_0, \mu_1, \sigma_0^2, \sigma_1^2\}$
- It is assumed that $\sigma_1^2 > \sigma_0^2$
- Time-homogeneous parameter set $\boldsymbol{\lambda} = \{\mathbf{A}, \boldsymbol{\pi}, \mathcal{B}\}$

Baum-Welch Algorithm

Define the posterior probability of the hidden state variable as

$$\gamma_t(i) = \Pr(s_t = i | \mathbf{x}_{1:T}, \boldsymbol{\lambda})$$

Once the HMM parameter set $\boldsymbol{\lambda}$ is known, the anomaly detection algorithm can be formulated as

$$\hat{s}_t = \operatorname{argmax}_{i \in \{0,1\}} \gamma_t(i).$$

Baum-Welch algorithm (an expectation-maximization algorithm) can learn parameter set, $\boldsymbol{\lambda}$, by maximizing the log-likelihood function

$$\ell = \log p(\mathbf{x}_{1:T} | \boldsymbol{\lambda})$$

Baum-Welch Algorithm

Baum-Welch algorithm involves the iterative calculation of a forward probability, $\alpha_t(i)$, and a backward probability, $\beta_t(i)$, defined as

$$\alpha_t(i) = \begin{cases} \pi_i p(x_1 | s_1 = i, \boldsymbol{\lambda}), & t = 1, \\ \Pr(\mathbf{x}_{1:t}, s_t = i | \boldsymbol{\lambda}), & 2 \leq t \leq T \end{cases}$$
$$\beta_t(i) = \begin{cases} p(\mathbf{x}_{(t+1):T} | s_t = i, \boldsymbol{\lambda}), & 1 \leq t \leq T - 1, \\ 1, & t = T. \end{cases}$$

The forward and backward probabilities can be iteratively updated as

$$\alpha_t(i) = \sum_{j \in \{0,1\}} \alpha_{t-1}(j) a_{ji} p(x_t | s_t = i, \boldsymbol{\lambda}), \quad 2 \leq t \leq T$$
$$\beta_t(i) = \sum_{j \in \{0,1\}} a_{ij} p(x_{t+1} | s_{t+1} = j, \boldsymbol{\lambda}) \beta_{t+1}(j),$$

for $1 \leq t \leq T - 1$.

Baum-Welch Implementation Challenges

- $\alpha_t(i)$ and $\beta_t(i)$ get small (mathematical underflow) for increasing T
- The 'log-sum-exponent' trick gives approximate solution with increasing time complexity
- We adopt the variable scaling approach as proposed in [Rab89] to achieve mathematically accurate and robust estimation
- [Rab89] is ambiguous while defining the scaled probabilities
- We try to give the physical meaning and better understanding of the scaled variables

Modified Baum-Welch Algorithm

Define a modified forward probability variable, $\tilde{\alpha}_t(i)$, as

$$\tilde{\alpha}_t(i) = \begin{cases} \Pr(\mathbf{x}_1, s_1 = i | \boldsymbol{\lambda}), & t = 1 \\ \Pr(\mathbf{x}_t, s_t = i | \mathbf{x}_{1:t-1}, \boldsymbol{\lambda}), & t > 1 \end{cases} \quad (1)$$

The variable $\tilde{\alpha}_t(i)$ can be interpreted as a scaled version of $\alpha_t(i)$ as

$$\tilde{\alpha}_t(i) = C_{t-1} \alpha_t(i) \quad (2)$$

where the scaling variable C_t is defined as follows

$$C_t^{-1} = \begin{cases} 1, & t = 0, \\ p(\mathbf{x}_{1:t} | \boldsymbol{\lambda}) = \alpha_t(0) + \alpha_t(1), & t \geq 1 \end{cases}$$

Lemma 1

Lemma

The scaled forward probability variable, $\tilde{\alpha}_t(i)$, can be calculated in an iterative manner as

$$\tilde{\alpha}_t(i) = c_{t-1} \sum_{j \in \{0,1\}} \tilde{\alpha}_{t-1}(j) \cdot a_{ji} \cdot p(x_t | s_t = i, \lambda),$$

for $2 \leq t \leq T$ and $\tilde{\alpha}_1(i) = \alpha_1(i)$, and c_t is defined as

$$c_t^{-1} = p(x_t | \mathbf{x}_{1:t-1}, \lambda) = \tilde{\alpha}_t(0) + \tilde{\alpha}_t(1)$$

Lemma 2

Lemma

The scaled backward probability variable, $\tilde{\beta}_t(i)$, can be calculated in an iterative manner as

$$\tilde{\beta}_t(i) = c_{t+1} \sum_{j \in \{0,1\}} a_{ij} p(x_{t+1} | s_{t+1} = j, \lambda) \tilde{\beta}_{t+1}(j) \quad (3)$$

where $1 \leq t \leq T - 1$ and $\tilde{\beta}_T(i) = \beta_T(i) = 1$.

Likelihood Function

Based on the definitions of C_t , D_t , and c_t , we have the following relationship among them

$$C_t = c_1 c_2 \cdots c_t = \prod_{i=1}^t c_i = c_t C_{t-1}$$
$$D_t = c_t c_{t+1} \cdots c_T = \prod_{i=t}^T c_i = c_t D_{t+1}$$

The likelihood function can then be calculated as

$$p(\mathbf{x}_{1:T} | \boldsymbol{\lambda}) = C_T^{-1} = \frac{c_t}{C_t D_t} = C_t^{-1} D_{t+1}^{-1} = c_t^{-1} C_{t-1}^{-1} D_{t+1}^{-1}$$

Posterior Probability

The posterior probability $\gamma_t(i)$ is redefined as

$$\begin{aligned}\gamma_t(i) &= \frac{p(\mathbf{x}_{1:T}, s_t = i | \boldsymbol{\lambda})}{p(\mathbf{x}_{1:T} | \boldsymbol{\lambda})} \\ &= \frac{\Pr(\mathbf{x}_{1:t}, s_t = i | \boldsymbol{\lambda}) \cdot p(\mathbf{x}_{(t+1):T} | s_t = i, \boldsymbol{\lambda})}{p(\mathbf{x}_{1:T} | \boldsymbol{\lambda})} \\ &= \alpha_t(i) \beta_t(i) C_{t-1} D_{t+1} c_t \\ &= c_t \tilde{\alpha}_t(i) \tilde{\beta}_t(i)\end{aligned}$$

- The relationship $\gamma_t(0) + \gamma_t(1) = 1$ can be used for sanity check
- In HMM, $\mathbf{x}_{1:t}$ and $\mathbf{x}_{t+1:T}$ are conditionally independent given s_t

Baum-Welch Algorithm

Implementation of the Baum-Welch algorithm requires the definition of the following probability

$$\begin{aligned}\xi_t(i, j) &= \Pr(s_t = i, s_{t+1} = j | \mathbf{x}_{1:T}, \boldsymbol{\lambda}) \\ &= \frac{\alpha_t(i) \beta_{t+1}(j) a_{ij} p(x_{t+1} | s_{t+1} = j)}{p(\mathbf{x}_{1:T} | \boldsymbol{\lambda})} \\ &= c_t a_{ij} \tilde{\alpha}_t(i) \tilde{\beta}_t(j) p(x_{t+1} | s_{t+1} = j)\end{aligned}$$

Baum-Welch Algorithm

- Baum-Welch algorithm contains an expectation step (E-step) and a maximization step (M-step)
- In the E-step, the probabilities, $\tilde{\alpha}_t(i)$, $\tilde{\beta}_t(i)$, $\gamma_t(i)$, and $\xi_t(i, j)$, are calculated with the parameter set λ from the previous iteration
- In the M-step, the parameter set λ are updated by using the probabilities $\gamma_t(i)$ and $\xi_t(i, j)$ obtained in the E step

Baum-Welch Algorithm

- The parameter estimations [ZML16] performed at the M -step in each iteration are

$$\begin{aligned}\pi_i &= \gamma_1(i); \\ \mu_i &= \frac{\sum_{t=1}^T \gamma_t(i) x_t}{\sum_{t=1}^T \gamma_t(i)} \\ \sigma_i^2 &= \frac{\sum_{t=1}^T \gamma_t(i) (x_t - \mu_i)^2}{\sum_{t=1}^T \gamma_t(i)} \\ a_{ij} &= \frac{\sum_{t=1}^{T-1} \xi_t(i, j)}{\sum_{t=1}^{T-1} \gamma_t(i)} \quad \text{for } i, j \in \{0, 1\}.\end{aligned}$$

- It is important to note that the estimated parameters are calculated using $\tilde{\alpha}_t(i)$ and $\tilde{\beta}_t(i)$ instead of using α_t and β_t

Baum-Welch Initialization

- In the k -th iteration, we can update the log-likelihood function as

$$\ell(k) = -\log C_T.$$

- Parameters are initialized as

$$\pi_0 = \pi_1 = 0.5$$

$$\mathbf{A} = \frac{1}{2} \mathbf{I}_2$$

$$\mu_0 = 0$$

$$\mu_1 = 0$$

$$\sigma_0 = s$$

$$\sigma_1 = \frac{1}{2} \max(|\mathbf{x}_{1:T}|)$$

- $\bar{x} = \sum_{i=1}^T x_i$ and $s^2 = \frac{1}{T-1} \sum_{i=1}^T (x_i - \bar{x})^2$

Algorithm 2 Baum-Welch Algorithm

- 1: **Input:** Discrete time dataset $\mathbf{x}_{1:T}$;
 - 2: Initialize λ , set $t = 0$ and $\ell(0) = -\infty$;
 - 3: **do**
 - 4: $t \leftarrow t + 1$;
 - 5: Calculate $\tilde{\alpha}_t(i)$ and c_t ;
 - 6: Calculate $\tilde{\beta}_t(i)$;
 - 7: Calculate $\gamma_t(i)$ and $\xi_t(i, j)$;
 - 8: Update $\pi_i, \mu_i, \sigma_i^2, \mathbf{A}$, here $\sigma_1^2 > \sigma_0^2$;
 - 9: Calculate $\ell(k)$;
 - 10: **while** $\ell(t) - \ell(t - 1) > \epsilon$
 - 11: **Output:** $\lambda, \gamma_t(i)$, for $t = 1, \dots, T$ and $i \in \{0, 1\}$.
-

Window-based Anomaly Removal

- Given $\gamma_t(i)$, hidden state \hat{s}_t of each sample can be estimated

$$\gamma_t(i) = \Pr(s_t = i | \mathbf{x}_{1:T}, \boldsymbol{\lambda})$$

- Samples with estimated hidden state $\hat{s}_t = 1$ are labeled as anomalies
- PVP signals from each subjects are first divided into non-overlapping windows with w samples per window
- If the percent of corresponding anomalous residuals within a window exceeds a certain threshold ζ (e.g. $\zeta = 15\%$), then all samples within this window are discarded.

Window-based Anomaly Removal

Algorithm 3 Window-based Anomaly Removal

- 1: **Input:** PVP signal $\mathbf{y}_{1:T}$, window size w , threshold ζ ;
- 2: Infer $\hat{\mathbf{s}}_{1:T}$ using Algorithms 1 and 2.
- 3: **for** $n = 1$ to $\lfloor T/w \rfloor$ **do**
- 4: Calculate the percentage of anomaly samples in the n -th window

$$\tau_n = \frac{1}{w} \sum_{i=1}^w \hat{S}_{(n-1)w+i}$$

- 5: If $\tau_n \geq \zeta$, discard the n -th window;
 - 6: **end for**
 - 7: **Output:** Normal windows.
-

Anomaly Detection: Example

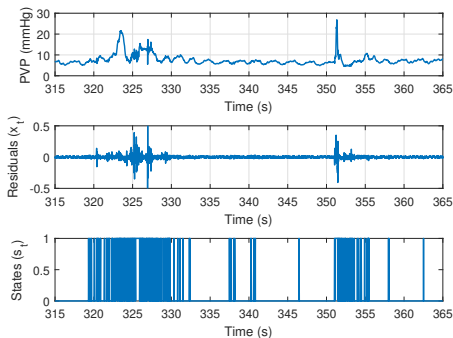


Figure 12: (Top) Exemplary PVP signal from Fig. 9; (middle) Prediction residual of the Kalman filter; and (bottom) estimated hidden states ($s_t = 0$: normal sample; $s_t = 1$: anomalous sample). Image: [HWB+20]

Distribution of Residuals $x(t)$

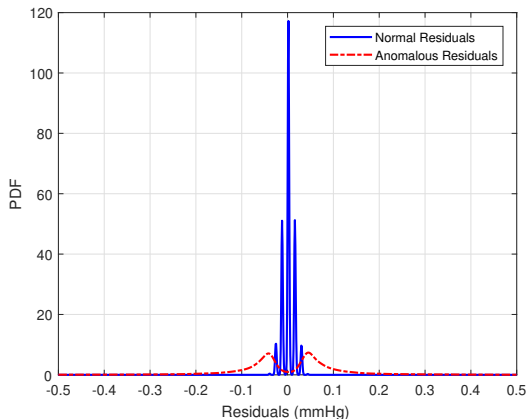


Figure 13: Empirical probability density function of prediction residuals (x_t) of patient 10. Image: [HWB+20]

Normal Windows: Examples

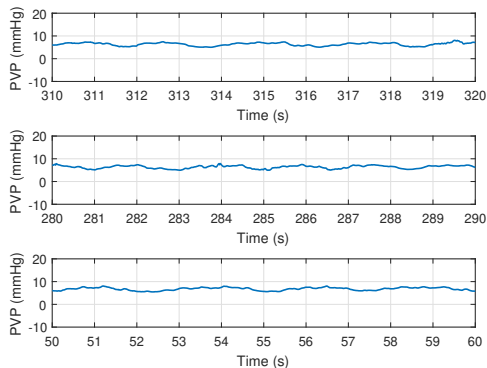


Figure 14: Example of normal windows inferred by the proposed model. Windows have a periodic structure and the amplitude does not change abruptly. Image: [HWB+20]

Anomalous Windows: Examples

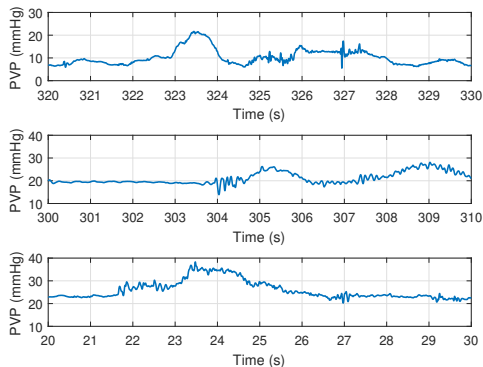


Figure 15: Example of anomalous windows inferred by the proposed model. In these windows, amplitude changes abruptly (10-15 mmHg higher than average) indicating random MNAs. Image: [HWB+20]

Anomaly Detection Results

Table 1: Number of normal and anomalous windows for 24 patients ($\zeta = 15\%$).

Patient	Status	Normal	Anomaly	Anomaly (%)
5	Hypo.	55	9	14.06
6	Hypo.	65	2	02.99
7	Hypo.	14	34	70.83
9	Euvo.	55	1	01.79
10	Euvo.	42	24	36.36
12	Euvo.	35	23	39.66
18	Hypo.	27	17	38.64
20	Euvo.	36	5	12.20
22	Hypo.	42	5	10.64
23	Euvo.	26	5	16.13
24	Euvo.	26	7	21.21
25	Hypo.	49	9	15.52
26	Hypo.	45	4	08.16
27	Euvo.	22	11	33.33
28	Euvo.	45	2	04.26
29	Hypo.	43	3	06.52
30	Euvo.	37	9	19.57
31	Euvo.	52	9	14.75
32	Hypo.	38	9	19.15
33	Euvo.	30	14	31.82
34	Hypo.	30	15	23.33
35	Euvo.	50	7	12.28
37	Euvo.	40	13	24.53
39	Euvo.	40	9	18.37

Patient Characteristics

Table 2: Characteristics of Euvolemic (Hydrated) and Hypovolemic (Dehydrated) patients

	Euvo.	Hypo.
Patients	14	10
Average weight (kg)	4.17	3.89
Minimum weight (kg)	2.76	2.72
Maximum weight (kg)	5.82	4.72
Std. deviation (kg)	0.78	0.70
Mean age (days)	38.1	40.1

Windows Used for Classification

Table 3: Windows in Euvolemic (Hydrated) and Hypovolemic (Dehydrated) patients

	Training Set			Testing Set		
	Euvo.	Hypo.	Total	Euvo.	Hypo.	Total
Raw Data	466	354	820	209	161	370
Manual	333	273	606	155	124	279
Algorithm in [BSM+19]	372	306	678	172	139	311
Proposed Algorithm	371	281	652	165	127	292

Table 4: Testing Classification Results

Parameter	Raw Data	Manual	Algorithm in [BSM+19]	Proposed Algorithm
True Positive Rate	45.96%	69.35%	63.31%	71.65%
True Negative Rate	76.08%	77.42%	79.65%	81.21%
Precision	59.68%	73.83%	71.54%	74.60%
F1 Score	52.00%	75.41%	67.18%	73.09%
Accuracy	62.97%	71.07%	72.35%	77.05%
Windows used	100%	70.20%	84.05%	78.92%

Future Works

- Based on a dataset collected during Summer 2020
 - Multiple signals were collected from four pigs
 - We have developed an analytical model (paper under review)
 - Try to distinguish different anesthetic levels using the dataset
- Using public dataset to understand similar scenarios
 - Exploring other non-invasive signals like PPG
 - Apply deep learning and methods already developed where possible

List of Publications (First Author Journals)

- **M. A. Hayat**, Jingxian Wu, et al. "Unsupervised Bayesian learning for rice panicle segmentation with UAV images." *Plant methods*, 16.1 (2020): 1-13.
- ✓ **M. A. Hayat**, Jingxian Wu, et al. "Unsupervised anomaly detection in peripheral venous pressure signals with hidden Markov models." *Biomedical Signal Processing and Control*, 62 (2020): 102126.
- **M. A. Hayat**, George Stein, et al. "Self-supervised representation learning for astronomical images." *The Astrophysical Journal Letters*, 911.2 (2021): L33.
- **M. A. Hayat**, Jingxian Wu, et al. "Modeling Peripheral Arterial and Venous Pressure Signals with Integral Pulse Frequency Modulation," *IEEE Transactions on Biomedical Engineering*. [Under Review]

List of Publications

- ✓ P. C. Bonasso, K. W. Sexton, **M. A. Hayat**, et al. “Venous physiology predicts dehydration in the pediatric population.” *Journal of Surgical Research*, 238 (2019): 232-239.
- P. C. Bonasso, K. W. Sexton, S. C. Mehl, M. S. Golinko, **M. A. Hayat**, et al. “Lessons learned measuring peripheral venous pressure waveforms in an anesthetized pediatric population.” *Biomedical Physics & Engineering Express* 5.3 (2019): 035020.
- A. Z. Al-Alawi, K. R. Henry, L. D. Crimmins, P. C. Bonasso, **M. A. Hayat**, et al. “Anesthetics affect peripheral venous pressure waveforms and the cross-talk with arterial pressure.” *Journal of clinical monitoring and computing* 36.1 (2022): 147-159.
- L. D. Crimmins-Pierce, G. P. Bonvillain, K. R. Henry, **M. A. Hayat**, et al. “Critical Information from High Fidelity Arterial and Venous Pressure Waveforms During Anesthesia and Hemorrhage.” *Cardiovascular Engineering and Technology* (2022): 1-13.

- Developed a supervised Gaussian mixture model based rice panicle segmentation algorithm using Markov chain Monte Carlo method
- Unlike deep learning models, it can work on smaller dataset and images with different sizes

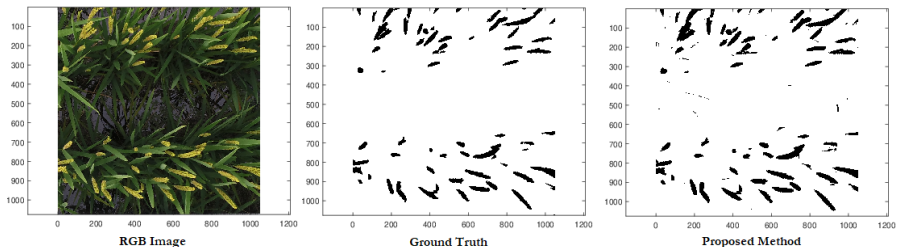


Figure 16: Panicle segmentation

- Nokia Bell Labs (Summer 2019)
 - OCT image processing using deep learning
- Lawrence Berkeley National Lab (Summer 2020)
 - Self-supervised learning of cosmological image
 - 1 Journal article
 - 1 NeurIPS workshop paper
 - Website and media coverage
- Amazon Web Services - AWS (Summer 2021)
 - Amazon Lookout for Metrics, an anomaly detection service
 - Applying deep learning based forecasting algorithm on anomaly detection



NEWS & EVENTS

News

[Archived News](#)

[Awards](#)

[Calendar](#)

[Alvarez Seminar Series](#)

[InTheLoop](#)

Self-Supervised ML Adds Depth, Breadth & Speed to Sky Surveys

In providing a general map of a particular region, sky surveys are also one of the largest data generators in science

MAY 11, 2021

By Kathy Kincade

Contact: cscotts@lbl.gov

Figure 17: Media coverage (source)

References

- [BSM+19] P. C. Bonasso *et al.*, “Lessons learned measuring peripheral venous pressure waveforms in an anesthetized pediatric population,” *Biomedical Physics & Engineering Express*, vol. 5, no. 3, p. 035 020, 2019.
- [HEY+06] N. Hadimioglu, Z. Ertug, A. Yegin, S. Sanli, A. Gurkan, and A. Demirbas, “Correlation of peripheral venous pressure and central venous pressure in kidney recipients,” in *Transplantation proceedings*, Elsevier, vol. 38, 2006, pp. 440–442.
- [HWB+20] M. A. Hayat *et al.*, “Unsupervised anomaly detection in peripheral venous pressure signals with hidden markov models,” *Biomedical Signal Processing and Control*, vol. 62, p. 102 126, 2020.
- [PA10] G. Petris and R. An, “An r package for dynamic linear models,” *Journal of Statistical Software*, vol. 36, no. 12, pp. 1–16, 2010.
- [Rab89] L. R. Rabiner, “A tutorial on hidden markov models and selected applications in speech recognition,” *Proceedings of the IEEE*, vol. 77, no. 2, pp. 257–286, 1989.
- [WMB04] J. E. Wathen, T. MacKenzie, and J. P. Bothner, “Usefulness of the serum electrolyte panel in the management of pediatric dehydration treated with intravenously administered fluids,” *Pediatrics*, vol. 114, no. 5, pp. 1227–1234, 2004.
- [ZML16] W. Zucchini, I. L. MacDonald, and R. Langrock, *Hidden Markov models for time series: an introduction using R*. Chapman and Hall/CRC, 2016.

Thank you for your patience!

- Questions?
- Feedback/Suggestions?