Deep Computational Phenotyping
Discovering latent representations of illness from clinical time series

Dave Kale*¹,², Zhengping Che*¹
Wenzhe Li¹, M. Taha Bahadori¹, Yan Liu¹
with thanks to Randall Wetzel²

¹ University of Southern California, Computer Science
² Laura P. and Leland K. Whittier VPICU, Children’s Hospital LA
* Equal contributions.

August 12, 2015
1 Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2 CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3 Exploiting structure in clinical time series
   Relationships between diseases ⇒ prior-based regularization
   Temporal smoothness ⇒ incremental training

4 Interpretation of learned phenotypes

5 Conclusion

6 References
Outline

1 Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2 CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3 Exploiting structure in clinical time series
   Relationships between diseases ⇒ prior-based regularization
   Temporal smoothness ⇒ incremental training

4 Interpretation of learned phenotypes

5 Conclusion

6 References
### Pediatric circulatory shock phenotypes [1, Table 1]

<table>
<thead>
<tr>
<th>Pathophysiology</th>
<th>Signs and Symptoms</th>
<th>Treatment</th>
</tr>
</thead>
</table>
| ↑ CO, ↓ SVR     | ↑ HR, ↓ BP, ↑ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of BP with dopamine or norepinephrine |
| ↓ CO, ↑ SVR     | ↑ HR, normal to ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of CO with dopamine or epinephrine |
| ↓ CO, ↓ SVR     | ↑ HR, ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of BP and CO with dopamine or epinephrine |
### Pediatric circulatory shock phenotypes [1, Table 1]

<table>
<thead>
<tr>
<th>Pathophysiology</th>
<th>Signs and Symptoms</th>
<th>Treatment</th>
</tr>
</thead>
</table>
| ↑ CO, ↓ SVR     | ↑ HR, ↓ BP, ↑ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of BP with dopamine or norepinephrine |
| ↓ CO, ↑ SVR     | ↑ HR, normal to ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of CO with dopamine or epinephrine |
| ↓ CO, ↓ SVR     | ↑ HR, ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of BP and CO with dopamine or epinephrine |

+ Parsimonious, easy to remember.
+ Abstract, flexible.
# Pediatric circulatory shock phenotypes [1, Table 1]

<table>
<thead>
<tr>
<th>Pathophysiology</th>
<th>Signs and Symptoms</th>
<th>Treatment</th>
</tr>
</thead>
<tbody>
<tr>
<td>↑ CO, ↓ SVR</td>
<td>↑ HR, ↓ BP, ↑ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema</td>
<td>Repeat boluses of 20 mL/kg crystalloid; may need &gt;60 mL/kg in first hour. Consider colloid if poor response to crystalloid. Pharmacologic support of BP with dopamine or norepinephrine.</td>
</tr>
<tr>
<td>↓ CO, ↑ SVR</td>
<td>↑ HR, normal to ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema</td>
<td>Repeat boluses of 20 mL/kg crystalloid; may need &gt;60 mL/kg in first hour. Consider colloid if poor response to crystalloid. Pharmacologic support of CO with dopamine or epinephrine.</td>
</tr>
<tr>
<td>↓ CO, ↓ SVR</td>
<td>↑ HR, ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema</td>
<td>Repeat boluses of 20 mL/kg crystalloid; may need &gt;60 mL/kg in first hour. Consider colloid if poor response to crystalloid. Pharmacologic support of BP and CO with dopamine or epinephrine.</td>
</tr>
</tbody>
</table>

+ Parsimonious, easy to remember.
+ Abstract, flexible.

- Imprecise, overlap, ignore other data.
- Difficult to automate.
### Pediatric circulatory shock phenotypes [1, Table 1]

<table>
<thead>
<tr>
<th>Pathophysiology</th>
<th>Signs and Symptoms</th>
<th>Treatment</th>
</tr>
</thead>
</table>
| ↑ CO, ↓ SVR     | ↑ HR, ↓ BP, ↑ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of BP with dopamine or norepinephrine |
| ↓ CO, ↑ SVR     | ↑ HR, normal to ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of CO with dopamine or epinephrine |
| ↓ CO, ↓ SVR     | ↑ HR, ↓ BP, ↓ pulses, delayed CR, hyperpnea, MS changes, third-spacing, edema | Repeat boluses of 20 mL/kg crystalloid; may need >60 mL/kg in first hour  
Consider colloid if poor response to crystalloid  
Pharmacologic support of BP and CO with dopamine or epinephrine |

+ Parsimonious, easy to remember.  
+ Abstract, flexible.  

**Traditional:** targeted to human caregivers, discovery driven by anecdote [2].  
− Imprecise, overlap, ignore other data.  
− Difficult to automate.
Computational phenotyping (CP) [2]

Digital health data (e.g., EHR)

\[
\begin{align*}
    h_1 &= \frac{1}{1 + e^{- (W_1 x + b_1)}} \\
    h_2 &= \frac{1}{1 + e^{- (W_2 h_1 + b_2)}} \\
    h_3 &= \frac{1}{1 + e^{- (W_3 h_2 + b_3)}} \\
    \hat{y} &= \frac{1}{1 + e^{- (W_4 h_3 + b_4)}}
\end{align*}
\]

Feed-forward computation of activations

Learning via back-propagation of gradient of cost function

\[
C(W, b) = \frac{1}{2} |f_{W, b}(x) - y|^2 \sum_{x, y}
\]

Cost function:

\[
W_1(ij) = W_1(ij) - \alpha \frac{\partial}{\partial W_1(ij)} C(W, b)
\]

input: x

Data-driven representations of disease

0.97 0.11 0.43 0.88 0.67 0.52 0.18 0.92 0.89 0.08

Text, codes data + non-negative tensor factorization.


Longitudinal events time series + matrix factorization.


Longitudinal labs time series + Bayesian cluster modeling.

Computational phenotyping (CP) \[2\]

Digital health data (e.g., EHR)

\[
\begin{align*}
  h_1 &= \frac{1}{1 + e^{- (W_1 x + b_1)}} \\
  h_2 &= \frac{1}{1 + e^{- (W_2 h_1 + b_2)}} \\
  h_3 &= \frac{1}{1 + e^{- (W_3 h_2 + b_3)}} \\
  \hat{y} &= \frac{1}{1 + e^{- (W_4 h_3 + b_4)}}
\end{align*}
\]

Feed-forward computation

\[
C(W, b) = \frac{1}{2} \sum_{(x,y)} |f_{W,b}(x) - y|^2
\]

Cost function:

\[
W_1(ij) = W_1(ij) - \alpha \frac{\partial}{\partial W_1(ij)} C(W, b)
\]
Computational phenotyping (CP) [2]

Digital health data (e.g., EHR)

Computational methods

Data-driven representations of disease

\[ h_1 = \frac{1}{1 + e^{- (W_1 x + b_1)}} \]
\[ h_2 = \frac{1}{1 + e^{- (W_2 h_1 + b_2)}} \]
\[ h_3 = \frac{1}{1 + e^{- (W_3 h_2 + b_3)}} \]
\[ \hat{y} = \frac{1}{1 + e^{- (W_4 h_3 + b_4)}} \]

\[ C(W, b) = \frac{1}{2} \sum_{(x,y)} |f_{W,b}(x) - y|^2 \]

Learning via backpropagation of gradient of cost function

Cost function:

\[ W_1(ij) = W_1(ij) - \alpha \frac{\partial}{\partial W_1(ij)} C(W, b) \]

Input: \( x \)
Computational phenotyping (CP) [2]

Digital health data (e.g., EHR)

Computational methods

Data-driven representations of disease

\[ h_1 = \frac{1}{1 + e^{- (W_1 x + b_1)}} \]
\[ h_2 = \frac{1}{1 + e^{- (W_2 h_1 + b_2)}} \]
\[ h_3 = \frac{1}{1 + e^{- (W_3 h_2 + b_3)}} \]
\[ \hat{y} = \frac{1}{1 + e^{- (W_4 h_3 + b_4)}} \]

Feedforward computation of activations

\[ C(W, b) = \frac{1}{2} \sum_{(x,y)} |f_{W,b}(x) - y|^2 \]

Cost function:

\[ W_1(ij) = W_1(ij) - \alpha \frac{\partial}{\partial W_1(ij)} C(W, b) \]

Input: \( x \)


Computational phenotyping (CP) [2]

Our setting: learning critical illness phenotypes from multivariate ICU time series.

Deformable motifs (Saria, et al. [6])

Bayesian clustering (Marlin, et al. [7])

Autoencoders (Lasko, et al. [2])
Biomarkers vs. features

**Medicine**: phenotypes, biomarkers [8]

1. Measurable attributes of patient/disease.
2. Independent of other biomarkers.
3. Separate patients into meaningful groups.
4. Improve outcome prediction, risk assessment.
5. Clinically plausible, interpretable.
**Biomarkers vs. features**

**Medicine:** *phenotypes, biomarkers* [8]
1. Measurable attributes of patient/disease.
2. Independent of other biomarkers.
3. Separate patients into meaningful groups.
4. Improve outcome prediction, risk assessment.
5. Clinically plausible, interpretable.

**Machine learning:** *features, representations* [9]
1. Measurable properties of objects.
2. Independent, disentangle factors of variation.
3. Form natural clusters.
4. Useful for discriminative, predictive tasks.
5. Interpretable, provide insight into problem.
Outline

1 Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2 CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3 Exploiting structure in clinical time series
   Relationships between diseases \(\Rightarrow\) prior-based regularization
   Temporal smoothness \(\Rightarrow\) incremental training

4 Interpretation of learned phenotypes

5 Conclusion

6 References
Learning critical care phenotypes from time series

Our setting: learning critical illness phenotypes from multivariate ICU time series.

Primary goal: interpretable latent space representations of critical care phenotypes.
Secondary goal: classification of critical illness phenotypes.

Kale (USC/VPICU)
Deep Computational Phenotyping
August 12, 2015
Our setting: learning critical illness phenotypes from multivariate ICU time series.

Respiratory disease (ICD-9 category 5)

Primary goal: interpretable latent space representations of critical care phenotypes.

Secondary goal: classification of critical illness phenotypes.
Our setting: learning critical illness phenotypes from multivariate ICU time series.

**Primary goal:**
interpretable latent space representations of critical care phenotypes.

**Secondary goal:**
classification of critical illness phenotypes.

---

**Respiratory disease**
(ICS-9 category 5)

**Septic shock**
(ICS-9: 785.52)

---

- DBP
- SBP
- CRR
- ETCO2
- FIO2
- TGCS
- Gluc
- HR
- pH
- RR
- SAO2
- Temp
- UO
Learning critical care phenotypes from time series

Our setting: learning critical illness phenotypes from multivariate ICU time series.

**Respiratory disease**  
*(ICD-9 category 5)*

**Septic shock**  
*(ICD-9: 785.52)*

**Primary goal:** interpretable latent space representations of critical care phenotypes.  
(i.e., features that detect characteristic patterns)

**Secondary goal:** classification of critical illness phenotypes.
Deep learning for time series: *sliding window approach*

\[ \hat{y} = \max_y \frac{1}{N} \sum_t P(\hat{y} \mid X_t) \]

Full Time Series Classification

Classification

Feature extraction

- Can use multi-task neural net (MTNNet) [10] to classify multiple phenotypes
- Can train multiple NNets for different window sizes
Deep learning for time series: *sliding window approach*

- Can use multi-task neural net (MTNNet) [10] to classify multiple phenotypes
- Can train multiple NNets for different window sizes $T$
Clinical data sets

8500 multivariate time series from CHLA PICU (*PICU*) [7]:

- All > 24 hours long.
- Sampled once per hour (*after preprocessing*).
- 13 variables: vitals, labs, outputs, assessments.
- Labels: 67 groups of ICD-9 codes, 19 standard ICD-9 categories.

8000 multivariate time series from PhysioNet Challenge 2012 (*PC2012*):

- 48 hours long (not full episodes in all cases).
- Sampled once per hour (*after preprocessing*).
- 33 variables: vitals, labs, outputs, assessments.
- Labels: in-hospital mortality, LOS > 3, cardiac, post-surgical

* Age correction (PICU only), resampling, imputation, rescaling, etc.
Clinical data sets

8500 multivariate time series from CHLA PICU (PICU) [7]:

- All > 24 hours long.
- Sampled once per hour (after preprocessing*).
- 13 variables: vitals, labs, outputs, assessments.
- Labels: 67 groups of ICD-9 codes, 19 standard ICD-9 categories.

8000 multivariate time series from PhysioNet Challenge 2012† (PC2012):

- 48 hours long (not full episodes in all cases).
- Sampled once per hour (after preprocessing*).
- 33 variables: vitals, labs, outputs, assessments.
- Labels: in-hospital mortality, LOS > 3, cardiac, post-surgical

* Age correction (PICU only), resampling, imputation, rescaling, etc.
† http://physionet.org/challenge/2012/
General experimental setup

1. **Data preparation**
   - Generate 5-10 random training/validation/test splits of *episodes*.
   - Extract all *subsequences* of length $T$ from full time series.

2. **Model architecture, training details**
   - 3-5 layers, fully connected, sigmoid hidden units.
   - Unsupervised pretraining with stochastic denoising autoencoders.
   - Supervised training (with early stopping) as (MT)NNet.

3. **Quantitative evaluation**
   - Measure per-subsequence, per-episode classification performance.
   - AUROC per label/category, for all labels/categories, for all outputs.

4. **Qualitative evaluation (i.e., feature visualization)**
Outline

1. Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2. CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3. Exploiting structure in clinical time series
   Relationships between diseases ⇒ prior-based regularization
   Temporal smoothness ⇒ incremental training

4. Interpretation of learned phenotypes

5. Conclusion

6. References
Exploiting structured domain knowledge

How do we handle sparsity in our labels?
Many diagnoses occur in < 1% of patients.
Exploiting structured domain knowledge

How do we handle sparsity in our labels?
Many diagnoses occur in <1% of patients.

How can we incorporate (structured) domain knowledge?
Ontologies (e.g., ICD-9 diagnostic codes) describe relationships between diseases.
Exploiting structured domain knowledge

How do we handle sparsity in our labels?
Many diagnoses occur in $< 1\%$ of patients.

How can we incorporate (structured) domain knowledge?
Ontologies (e.g., ICD-9 diagnostic codes) describe relationships between diseases.

Graph Laplacian prior

Assume:

- $K$ outputs (labels) with parameters $\{\beta_k\}_{k=1}^{K}$, $\beta_k \in \mathbb{R}^{D(L)}$
- Label similarity matrix $A \in \mathbb{R}^{K \times K}$ where $A_{ij} \in [0, 1]$.

Define Graph Laplacian matrix $L = C - A$ with $C$ a diagonal matrix $C_{kk} = \sum_{k'=1}^{K} A_{kk'}$, then

$$\text{tr}(\beta \top L \beta) = \sum_{1 \leq k, k' \leq K} A_{k,k'} \|\beta_k - \beta_{k'}\|_2^2$$

where $\text{tr}(\cdot)$ represents the trace operator.

New regularized loss function for supervised training of MTNNet:

$$L = -\frac{1}{N} \sum_{i=1}^{K} \sum_{k=1}^{K} \left[ y_{ik} \log \sigma(\beta \top k h_i) + (1 - y_{ik}) \log(1 - \sigma(\beta \top k h_i)) \right] + \rho \text{tr}(\beta \top L \beta)$$
Graph Laplacian prior

Assume:
- $K$ outputs (labels) with parameters $\{\beta_k\}_{k=1}^{K}$, $\beta_k \in \mathbb{R}^{D(L)}$
- Label similarity matrix $A \in \mathbb{R}^{K \times K}$ where $A_{ij} \in [0, 1]$.

Define Graph Laplacian matrix

$$L = C - A$$

with $C$ a diagonal matrix $C_{kk} = \sum_{k'=1}^{K} A_{kk'}$, then

$$\text{tr}(\beta^\top L \beta) = \sum_{1 \leq k, k' \leq K} A_{k,k'} \|\beta_k - \beta_{k'}\|_2^2$$

where $\text{tr}(\cdot)$ represents the trace operator.
Graph Laplacian prior

Assume:

- $K$ outputs (labels) with parameters $\{\beta_k\}_{k=1}^K$, $\beta_k \in \mathbb{R}^{D(L)}$
- Label similarity matrix $A \in \mathbb{R}^{K \times K}$ where $A_{ij} \in [0, 1]$.

Define Graph Laplacian matrix

$$L = C - A$$

with $C$ a diagonal matrix $C_{kk} = \sum_{k'=1}^K A_{kk'}$, then

$$\text{tr}(\beta^\top L \beta) = \sum_{1 \leq k, k' \leq K} A_{k,k'} \|\beta_k - \beta_{k'}\|_2^2$$

where $\text{tr}(\cdot)$ represents the trace operator.

New regularized loss function for supervised training of MTNNet:

$$L = - \sum_{i=1}^N \sum_{k=1}^K \left[ y_{ik} \log \sigma(\beta_k^\top h_i) + (1 - y_{ik}) \log(1 - \sigma(\beta_k^\top h_i)) \right] + \frac{\rho}{2} \text{tr}(\beta^\top L \beta)$$
Laplacian priors can incorporate arbitrary similarities

Tree-based priors [12]

Co-occurrence priors (i.e., co-morbidity)
Impact of priors on phenotype classification

**PICU data** (AUROC across 67 labels and 19 categories from ICD-9 codes)

<table>
<thead>
<tr>
<th>Tasks</th>
<th>No Prior</th>
<th>Co-Occurrence</th>
<th>ICD-9 Tree</th>
</tr>
</thead>
<tbody>
<tr>
<td>Subsequence</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>0.7079 ± 0.0089</td>
<td>0.7169 ± 0.0087</td>
<td>0.7143 ± 0.0066</td>
</tr>
<tr>
<td>Categories</td>
<td>0.6758 ± 0.0078</td>
<td>0.6804 ± 0.0109</td>
<td>0.6710 ± 0.0070</td>
</tr>
<tr>
<td>Labels</td>
<td>0.7148 ± 0.0114</td>
<td>0.7241 ± 0.0093</td>
<td>0.7237 ± 0.0081</td>
</tr>
<tr>
<td>Episode</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>All</td>
<td>0.7245 ± 0.0077</td>
<td>0.7348 ± 0.0064</td>
<td>0.7316 ± 0.0062</td>
</tr>
<tr>
<td>Categories</td>
<td>0.6952 ± 0.0106</td>
<td>0.7010 ± 0.0136</td>
<td>0.6902 ± 0.0118</td>
</tr>
<tr>
<td>Labels</td>
<td>0.7308 ± 0.0099</td>
<td>0.7414 ± 0.0064</td>
<td>0.7407 ± 0.0070</td>
</tr>
</tbody>
</table>

**Physionet Challenge 2012 data**

<table>
<thead>
<tr>
<th>Task</th>
<th>Indep. baseline</th>
<th>ML baseline</th>
<th>Co-Oc. Prior</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mortality</td>
<td>0.75 ± 0.01</td>
<td>0.80 ± 0.00</td>
<td>0.85 ± 0.00</td>
</tr>
<tr>
<td>LOS&lt;3</td>
<td>0.70 ± 0.01</td>
<td>0.75 ± 0.00</td>
<td>0.80 ± 0.00</td>
</tr>
<tr>
<td>Surgery</td>
<td>0.80 ± 0.00</td>
<td>0.85 ± 0.00</td>
<td>0.90 ± 0.00</td>
</tr>
<tr>
<td>Cardiac</td>
<td>0.85 ± 0.00</td>
<td>0.90 ± 0.00</td>
<td>0.95 ± 0.00</td>
</tr>
</tbody>
</table>

*Mortality:* in-hospital mortality; *LOS<3:* length-of-stay < 3 days; *Surgery:* patient received surgery; *Cardiac:* cardiac patient
How can we learn patterns of different lengths?
Requires training a series of neural nets with overlapping architectures. **Less training data for longer patterns.**
When training on subsequences, there are fewer longer subsequences.

**Solution:**
initialize larger network’s parameters using smaller network; exploit smoothness in data, structure in weights. [13]
Exploiting temporal smoothness of clinical time series

How can we learn patterns of different lengths?
Requires training a series of neural nets with overlapping architectures.
Less training data for longer patterns.
When training on subsequences, there are fewer longer subsequences.

Solution: initialize larger network's parameters using smaller network; exploit smoothness in data, structure in weights. [13]
Incremental training of neural nets for time series mining

Given NNet for $T$ patterns of $P$ variables (input size $D = TP$, weights $W$). Initialize training of NNet for $T' > T$ patterns (weights $W'$).

Adding $d$ inputs adds $d$ columns to $W$; $d^{(1)}$ features adds $d^{(1)}$ rows to $W$:

$$
N_1 \{ \begin{array}{c} h_1 \\ n \end{array} \} = f 
N_1 \{ \begin{array}{c} W_1 \\ \Delta W_{ne} \\ \Delta W_{en} \\ \Delta W_{nn} \\ b_1 \\ n \end{array} \} + \begin{array}{c} \Delta x \\ d \end{array} \Delta b \begin{array}{c} D \\ n \end{array}
$$

Construct $W'$ from $W$ as follows:

$$W' = \begin{bmatrix}
W = I_{D^{(1)}} W \\
\Delta W_{ne} = W K \\
\Delta W_{en} = R_{en} W C_{ne} \\
\Delta W_{nn} = R_{nn} W C_{nn}
\end{bmatrix}
$$

for input kernel similarity (e.g., covariance) matrix $K \in \mathbb{R}^{D \times d}$. Build matrices $R \bullet, C \bullet$ by sampling rows and columns from $I_D$. 
Incremental training of neural nets for time series mining

Given NNet for \( T \) patterns of \( P \) variables (input size \( D = TP \), weights \( W \)). Initialize training of NNet for \( T' > T \) patterns (weights \( W' \)).

Adding \( d \) inputs adds \( d \) columns to \( W \); \( d^{(1)} \) features adds \( d^{(1)} \) rows to \( W \):

\[
\begin{align*}
\Delta W_{ne} & \quad \Delta W_{en} & \quad \Delta W_{nn} \\
\Delta h & \quad \Delta x & \quad \Delta b
\end{align*}
\]

Construct \( W' \) from \( W \) as follows:

- \( W \): weight matrix from \( D \)-window net.
- \( \Delta W_{ne} \): similarity-weighted linear combinations of \( W \)’s columns.
- \( \Delta W_{en}, \Delta W_{nn} \): sample from \( W \) weights.
Incremental training of neural nets for time series mining

Given NNet for $T$ patterns of $P$ variables (input size $D = TP$, weights $W$).
Initialize training of NNet for $T' > T$ patterns (weights $W'$).

Adding $d$ inputs adds $d$ columns to $W$; $d^{(1)}$ features adds $d^{(1)}$ rows to $W$:

\[
\begin{align*}
N_1 \{ h_1 \} &= f \\
N_1 \{ \Delta h \} &= \Delta W_{ne} \quad \Delta W_{nn} \\
N_1 \{ x \} &= d \\
N_1 \{ \Delta x \} &= d \\
N_1 \{ b_1 \} &= N_1 \{ \Delta b \} \\
&= f + \Delta W_1 + \Delta W_{ne} + \Delta W_{nn} + b_1 + \Delta b
\end{align*}
\]

Intuitions:
- If training on sliding windows, longer subsequences contain shorter.
- Features exploit input regularity [13]; clinical time series often temporally smooth.
- Overlap, smoothness in data $\Rightarrow$ overlap, structure in weights.

Note: works for all layers.
Combining different window sizes improves classification:

<table>
<thead>
<tr>
<th>$T(s)$</th>
<th>AUC</th>
<th>Precision</th>
<th>Recall</th>
</tr>
</thead>
<tbody>
<tr>
<td>{8,12,16,20,24}*</td>
<td>0.8076 ± 0.0106</td>
<td>0.6596 ± 0.0198</td>
<td>0.7354 ± 0.0150</td>
</tr>
<tr>
<td>{8,12,16,20,24}</td>
<td>0.8014 ± 0.0087</td>
<td>0.6590 ± 0.0201</td>
<td>0.7204 ± 0.0187</td>
</tr>
<tr>
<td>{8,12,24}</td>
<td>0.7999 ± 0.0098</td>
<td>0.6597 ± 0.0182</td>
<td>0.7166 ± 0.0165</td>
</tr>
<tr>
<td>24</td>
<td>0.7898 ± 0.0202</td>
<td>0.6443 ± 0.0327</td>
<td>0.7025 ± 0.0160</td>
</tr>
<tr>
<td>12</td>
<td>0.7870 ± 0.0129</td>
<td>0.6349 ± 0.0118</td>
<td>0.7161 ± 0.0211</td>
</tr>
</tbody>
</table>

* Trained incrementally.
**Incremental vs. full training:** PICU data, Respiratory category

For $T = 12$: reduced training time, competitive classification.

<table>
<thead>
<tr>
<th>Method</th>
<th># Sup. Epochs</th>
<th>Total Training Time</th>
<th>Val. Error</th>
<th>Test Error</th>
</tr>
</thead>
<tbody>
<tr>
<td>Incremental</td>
<td>55.8</td>
<td>306.54</td>
<td>29.29%</td>
<td>29.38%</td>
</tr>
<tr>
<td>Full</td>
<td>23.6</td>
<td>471.74</td>
<td>32.23%</td>
<td>32.07%</td>
</tr>
</tbody>
</table>

---

![Graph showing performance comparison](chart.png)

- NNet
- NNetInc(Ft)
- NNetInc(FtOnly)
- NNetInc(Pt)
Incremental vs. full training: PICU data, multi-label

- Eliminates need unsupervised pretraining.
- Comparable supervised training time, classification performance.
- Interacts well with prior-based regularizers.

<table>
<thead>
<tr>
<th>Window Size</th>
<th>Full Training Time (min)</th>
<th>Prior + Full Training Time (min)</th>
<th>Prior + Inc. Training Time (min)</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>16</td>
<td>20</td>
<td>24</td>
</tr>
<tr>
<td>20</td>
<td>40</td>
<td>40</td>
<td>40</td>
</tr>
<tr>
<td>40</td>
<td>60</td>
<td>60</td>
<td>60</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>T</th>
<th>Level</th>
<th>Full</th>
<th>Inc</th>
</tr>
</thead>
<tbody>
<tr>
<td>16</td>
<td>Seq.</td>
<td>0.6556</td>
<td>0.6581</td>
</tr>
<tr>
<td></td>
<td>Ep.</td>
<td>0.6668</td>
<td>0.6744</td>
</tr>
<tr>
<td>20</td>
<td>Seq.</td>
<td>0.6674</td>
<td>0.6746</td>
</tr>
<tr>
<td></td>
<td>Ep.</td>
<td>0.6794</td>
<td>0.6944</td>
</tr>
<tr>
<td>24</td>
<td>Seq.</td>
<td>0.6946</td>
<td>0.7008</td>
</tr>
<tr>
<td></td>
<td>Ep.</td>
<td>0.7136</td>
<td>0.7171</td>
</tr>
</tbody>
</table>
Outline

1 Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2 CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3 Exploiting structure in clinical time series
   Relationships between diseases ⇒ prior-based regularization
   Temporal smoothness ⇒ incremental training

4 Interpretation of learned phenotypes

5 Conclusion

6 References
Interpretation and visualization of phenotypic features

To create interpretable visualizations of phenotypic features:

1. Choose (using, e.g., Lasso) most predictive features for phenotype.
2. Find inputs with highest aggregate (across those features) activation.
3. Plot mean (and standard deviation) trajectories.
Interpretation and visualization of phenotypic features

To create interpretable visualizations of phenotypic features:

1. Choose (using, e.g., Lasso) most predictive features for phenotype.
2. Find inputs with highest aggregate (across those features) activation.
3. Plot mean (and standard deviation) trajectories.

Alternative feature selection: **causal analysis via Pairwise LiNGAM (PL)** [14]:

\[
R = \frac{1}{N} \log L(h_j \rightarrow y_k) - \frac{1}{N} \log L(h_j \leftarrow y_k)
\]

\[
\begin{cases} 
    R > 0 & \text{if } h_j \rightarrow y_k \\
    R < 0 & \text{if } h_j \leftarrow y_k 
\end{cases}
\]
Interpretation and visualization of phenotypic features

To create interpretable visualizations of phenotypic features:

1. Choose (using, e.g., Lasso) most predictive features for phenotype.
2. Find inputs with highest aggregate (across those features) activation.
3. Plot mean (and standard deviation) trajectories.

Alternative feature selection: causal analysis via Pairwise LiNGAM (PL) [14]:

\[
R = \frac{1}{N} \log L(h_j \rightarrow y_k) - \frac{1}{N} \log L(h_j \leftarrow y_k)
\]

\[
\begin{cases} 
R > 0 & \text{if } h_j \rightarrow y_k \\
R < 0 & \text{if } h_j \leftarrow y_k 
\end{cases}
\]

Keep causal (via PL) + correlated (via Lasso) features:

We can consider the learned representations as causal hypotheses for phenotypes.
• Very irregular physiology, known symptoms of sepsis.
• Low Glasgow coma score indicates patient is unconscious.
• Elevated blood pressure and heart rate, depressed pH.
• Evidence of ventilation (elevated FIO2).
• Note elevated urine output; also correlated with urinary disorders.
Outline

1 Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2 CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3 Exploiting structure in clinical time series
   Relationships between diseases ⇒ prior-based regularization
   Temporal smoothness ⇒ incremental training

4 Interpretation of learned phenotypes

5 Conclusion

6 References
Conclusion

Deep learning a powerful tool for computational phenotyping!

- More thorough empirical investigation (Stanford clinic data, \(~1M\) patients)
- Alternative NNet architectures to eliminate need for preprocessing
- Multimodal N Nets to combine clinical time series with notes, events, etc.
- Model impact of treatments!

Dave Kale: http://www-scf.usc.edu/~dkale/
Yan Liu: http://www-bcf.usc.edu/~liu32/
Whitter Virtual PICU (VPICU): http://vpicu.org/

Meaningful Use of Complex Medical Data Symposium: http://mucmd.org/

Support: NSF research grants IIS-1134990 and IIS-1254206, Okawa Foundation Research Award, Alfred E. Mann Innovation in Engineering Fellowship, Laura P. and Leland K. Whittier Foundation grant, Children’s Hospital Los Angeles.

Thank you and fight on!
Outline

1 Background: why and how of computational phenotyping
   Phenotypes: representations of illness
   Computational phenotyping (CP)
   CP as representation learning

2 CP from clinical time series with deep learning
   Sliding window approach (e.g., subsequence mining)
   Data sets and general experimental setup

3 Exploiting structure in clinical time series
   Relationships between diseases ⇒ prior-based regularization
   Temporal smoothness ⇒ incremental training

4 Interpretation of learned phenotypes

5 Conclusion

6 References
References


References II


