Learning from Irregularly-Sampled Time Series

A Missing Data Perspective

Steven Cheng-Xian Li Benjamin M. Marlin

University of Massachusetts Amherst

Irregularly-Sampled Time Series

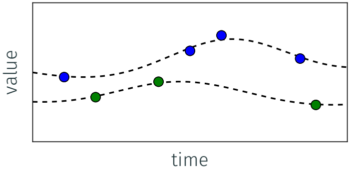
Irregularly-sampled time series:

Time series with non-uniform time intervals between successive measurements



Problem and Challenges

Problem: learning from a collection of *irregularly-sampled* time series within a common time interval

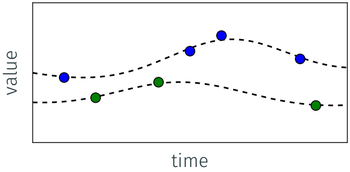


Challenges:

- Each time series is observed at *arbitrary time points*.
  - Different data cases may have different numbers of observations
  - Observed samples may not be aligned in time
  - Many real-world time series data are extremely sparse
- Most machine learning algorithms require data lying on fixed dimensional feature space

Problem and Challenges

Problem: learning from a collection of *irregularly-sampled* time series within a common time interval



Tasks:

- Learning the distribution of latent temporal processes
- Inferring the latent process associated with a time series
- Classification of time series

This can be transformed into a **missing data problem**.

Index Representation of Incomplete Data

Data defined on an **index set** $\mathcal{I}$:

- Examples:
  - Image: pixel coordinates
  - Time series: timestamps
- Complete data as a mapping: $\mathcal{I} \to \mathbb{R}$.

Index representation of an incomplete data case $(\mathbf{x}, \mathbf{t})$:

- $\mathbf{t} \equiv \{t_i\}_{i=1}^{|\mathbf{t}|} \subset \mathcal{I}$ are the indices of observed entries.
- x_i is the corresponding value observed at t_i.
- Applicable for both finite and continuous index set.

Index Representation of Incomplete Data

Data defined on an **index set** $\mathcal{I}$:

- Examples:
  - Image: pixel coordinates
  - Time series: timestamps
- Complete data as a mapping: $\mathcal{I} \to \mathbb{R}$.

Index representation of an incomplete data case $(\mathbf{x}, \mathbf{t})$:

- $\mathbf{t} \equiv \{t_i\}_{i=1}^{|\mathbf{t}|} \subset \mathcal{I}$ are the indices of observed entries.
- x_i is the corresponding value observed at t_i.
- Applicable for both finite and continuous index set.

Generative Process of Incomplete Data

Generative process for an incomplete case $(\mathbf{x}, \mathbf{t})$:

$f \sim p_\theta(f)$	complete data $f : \mathcal{I} \to \mathbb{R}$
$\mathbf{t} \sim p_\mathcal{I}(\mathbf{t}\|f)$	$\mathbf{t} \in 2^\mathcal{I}$ (subset of $\mathcal{I}$)
$\mathbf{x} = \left[f(t_i)\right]_{i=1}^{\|\mathbf{t}\|}$	values of f indexed at $\mathbf{t}$

Goal: learning the complete data distribution p_θ given the incomplete dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^n$

Generative Process of Incomplete Data

Generative process for an incomplete case $(\mathbf{x}, \mathbf{t})$:

$f \sim p_\theta(f)$	complete data $f : \mathcal{I} \to \mathbb{R}$
$\mathbf{t} \sim p_\mathcal{I}(\mathbf{t})$	*independence* between f and $\mathbf{t}$
$\mathbf{x} = \left[f(t_i)\right]_{i=1}^{\lvert\mathbf{t}\rvert}$	values of f indexed at $\mathbf{t}$

Goal: learning the complete data distribution p_θ given the incomplete dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^n$

Generative Process of Incomplete Data

Generative process for an incomplete case $(\mathbf{x}, \mathbf{t})$:

$f \sim p_\theta(f)$	complete data $f : \mathcal{I} \to \mathbb{R}$
$\mathbf{t} \sim p_\mathcal{I}(\mathbf{t})$	*independence* between f and $\mathbf{t}$
$\mathbf{x} = \left[f(t_i)\right]_{i=1}^{\lvert\mathbf{t}\rvert}$	values of f indexed at $\mathbf{t}$

Goal: learning the complete data distribution p_θ given the incomplete dataset $\mathcal{D} = \{(\mathbf{x}_i, \mathbf{t}_i)\}_{i=1}^n$

Encoder-Decoder Framework for Incomplete Data

Probabilistic latent variable model

Decoder:

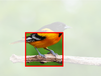
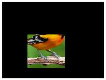
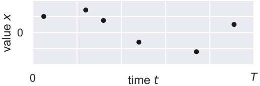
- Model the data generating process: $\mathbf{z} \sim p_z(\mathbf{z}),\ f = g_\theta(\mathbf{z})$
- Given $\mathbf{t} \sim p_\mathcal{I}$, the corresponding values are $g_\theta(\mathbf{z}, \mathbf{t}) \equiv \left[f(t_i)\right]_{i=1}^{|\mathbf{t}|}$.
- Note: our goal is to model g_θ, not $p_\mathcal{I}$.

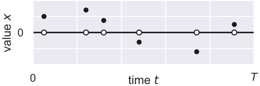
Encoder-Decoder Framework for Incomplete Data

Encoder (**stochastic**):

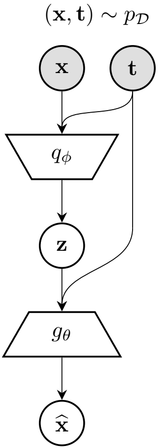
- Model the posterior distribution $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})$
- Functional form: $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t}) = q_\phi(\mathbf{z} \,|\, m(\mathbf{x},\mathbf{t}))$
  - Example: $q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t}) = \mathcal{N}(\mathbf{z}|\mu_\phi(\mathbf{v}), \Sigma_\phi(\mathbf{v}))$ with $\mathbf{v} = m(\mathbf{x},\mathbf{t})$.
- Different incomplete cases carry different levels of uncertainty.

Masking function $m(\mathbf{x},\mathbf{t})$:

- Replacing all missing entries by zero.
  - $m\Bigg($  $\Bigg) =$ 
  - $m\Bigg($ 

$\Bigg) =$ 

Partial Variational Autoencoder (P-VAE)



Generative process:

$$\mathbf{t} \sim p_{\mathcal{I}}(\mathbf{t})$$
$$\mathbf{z} \sim p(\mathbf{z})$$
$$f = g_\theta(\mathbf{z})$$
$$x_i \sim p(x_i | f(t_i)) \quad \text{(i.i.d. noise)}$$

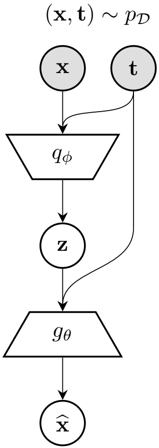
Example: $p(x_i | f(t_i)) = \mathcal{N}(x_i | f(t_i), \sigma^2)$

Joint distribution:

$$p(\mathbf{x}, \mathbf{t}) = \int p(\mathbf{z}) p_{\mathcal{I}}(\mathbf{t}) \prod_{i=1}^{|\mathbf{t}|} p_\theta(x_i | \mathbf{z}, t_i) d\mathbf{z}$$

$p_\theta(x_i | \mathbf{z}, t_i)$ is the shorthand for $p(x_i | f(t_i))$ with $f = g_\theta(\mathbf{z})$.

Partial Variational Autoencoder (P-VAE)



Variational lower bound for $\log p(\mathbf{x}, \mathbf{t})$:

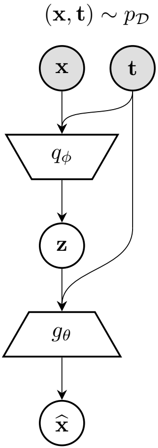
$$\int q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t}) \log \frac{p_z(\mathbf{z}) p_{\mathcal{I}}(\mathbf{t}) \prod_{i=1}^{|\mathbf{t}|} p_\theta(x_i|\mathbf{z}, t_i)}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})} d\mathbf{z}$$

Learning with gradients **without** $p_{\mathcal{I}}(\mathbf{t})$ **involved**:

$$\nabla_{\phi,\theta} \mathbb{E}_{\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})} \left[\log \frac{p_z(\mathbf{z}) \, \cancel{p_{\mathcal{I}}(\mathbf{t})} \prod_{i=1}^{|\mathbf{t}|} p_\theta(x_i|\mathbf{z}, t_i)}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})} \right]$$

Kingma & Welling. (2014). Auto-encoding variational bayes.
Ma, et al. (2018). Partial VAE for hybrid recommender system.

Partial Variational Autoencoder (P-VAE)



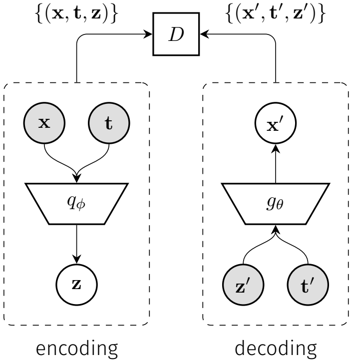
Conditional objective (lower bound for $\log p(\mathbf{x}|\mathbf{t})$):

$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\left[\log \frac{p_z(\mathbf{z})\prod_{i=1}^{|\mathbf{t}|} p_\theta(x_i|\mathbf{z}, t_i)}{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\right]$$

Related work:

- Partial VAE [Ma, et al., 2018]
- Neural processes [Garnelo, et al., 2018]
- MIWAE [Mattei & Frellsen, 2019]

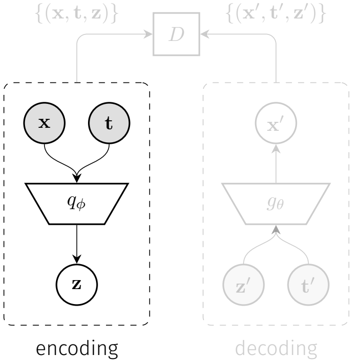
Partial Bidirectional GAN (P-BiGAN)



Li, Jiang, Marlin. (2019). MisGAN: Learning from Incomplete Data with GANs.
Donahue, et al. (2016). Adversarial feature learning (BiGAN).

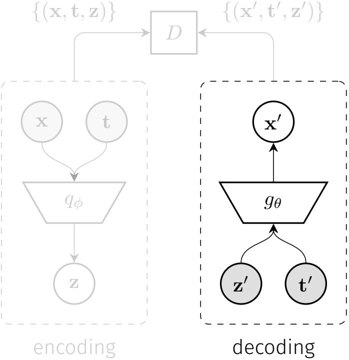
Partial Bidirectional GAN (P-BiGAN)

$(\mathbf{x}, \mathbf{t}) \sim p_{\mathcal{D}}$



Li, Jiang, Marlin. (2019). MisGAN: Learning from Incomplete Data with GANs.
Donahue, et al. (2016). Adversarial feature learning (BiGAN).

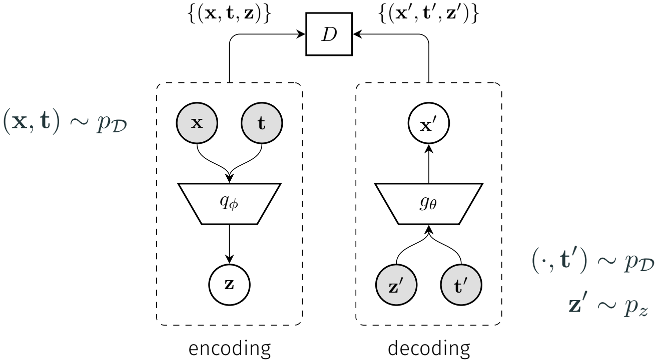
Partial Bidirectional GAN (P-BiGAN)



$$(\cdot, \mathbf{t}') \sim p_{\mathcal{D}}$$
$$\mathbf{z}' \sim p_z$$

Li, Jiang, Marlin. (2019). MisGAN: Learning from Incomplete Data with GANs.
Donahue, et al. (2016). Adversarial feature learning (BiGAN).

Partial Bidirectional GAN (P-BiGAN)

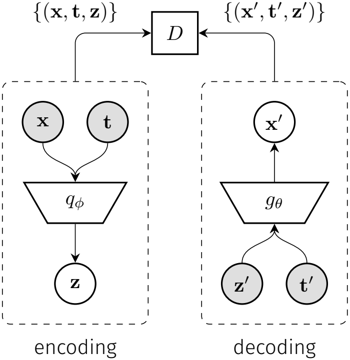


Discriminator: $D(m(\mathbf{x}, \mathbf{t}), \mathbf{z})$

Li, Jiang, Marlin. (2019). MisGAN: Learning from Incomplete Data with GANs.
Donahue, et al. (2016). Adversarial feature learning (BiGAN).

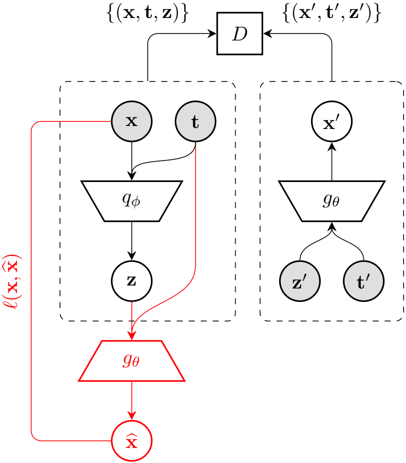
Invertibility Property of P-BiGAN

Theorem:
For $(\mathbf{x}, \mathbf{t})$ with non-zero probability, if $\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})$ then $g_\theta(\mathbf{z}, \mathbf{t}) = \mathbf{x}$.

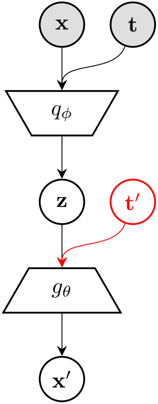


$g_\theta(\mathbf{z}, \mathbf{t})$ is the shorthand notation for $[f(t_i)]_{i=1}^{|\mathbf{t}|}$ with $f = g_\theta(\mathbf{z})$.

Autoencoding Regularization for P-BiGAN



Missing Data Imputation



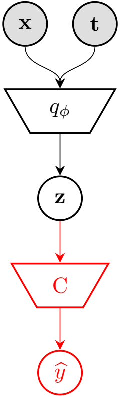
Imputation:

$$p(\mathbf{x}'|\mathbf{t}', \mathbf{x}, \mathbf{t}) = \mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\left[p_\theta(\mathbf{x}'|\mathbf{z}, \mathbf{t}')\right]$$

Sampling:

$$\mathbf{z} \sim q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})$$
$$f = g_\theta(\mathbf{z})$$
$$\mathbf{x}' = [f(t'_i)]_{i=1}^{|\mathbf{t}'|}$$

Supervised Learning: Classification



Adding classification term to objective:

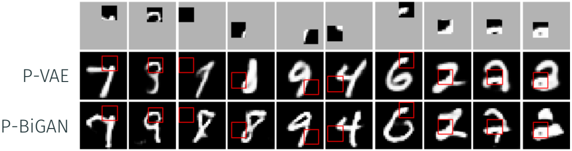
$$\mathbb{E}_{\mathbf{z}\sim q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\left[\log \frac{p_z(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z},\mathbf{t})p(y|\mathbf{z})}{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\right]$$

$$= \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\left[\log \frac{p_z(\mathbf{z})p_\theta(\mathbf{x}|\mathbf{z},\mathbf{t})}{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}\right]}_{\text{regularization}} + \underbrace{\mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}[\log p(y|\mathbf{z})]}_{\text{classification}}$$

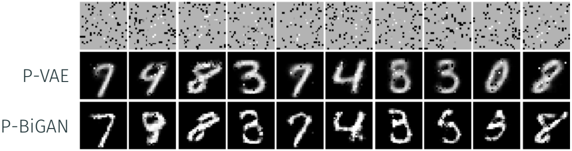
Prediction:

$$\widehat{y} = \underset{y}{\operatorname{argmax}}\, \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})}[\log p(y|\mathbf{z})]$$

MNIST Completion

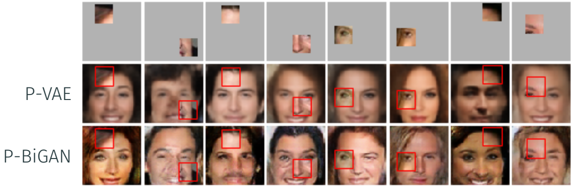


square observation with 90% missing

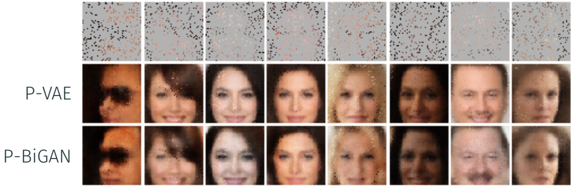


independent dropout with 90% missing

CelebA Completion



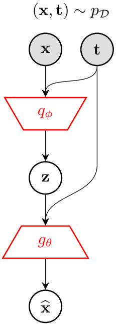
square observation with 90% missing



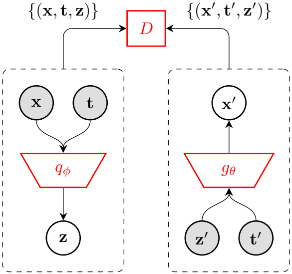
independent dropout with 90% missing

Architecture for Irregularly-Sampled Time Series

How to construct *decoder, encoder and discriminator* for continuous index set, e.g., time series with $\mathcal{I} = [0, T]$?



P-VAE



P-BiGAN

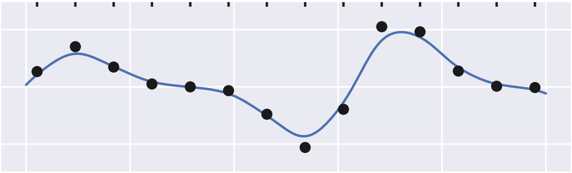
Decoder for Continuous Time Series

Generative process for time series:

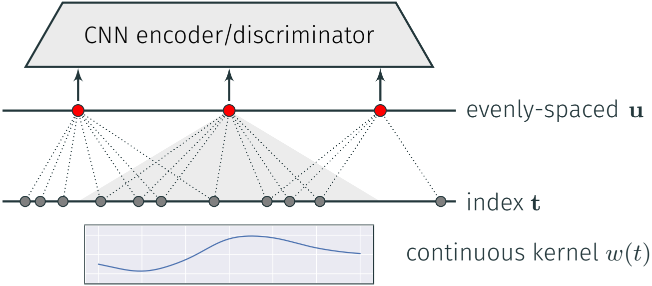
$$\mathbf{z} \sim p_z(\mathbf{z})$$

$$\mathbf{v} = \text{CNN}_\theta(\mathbf{z})$$ values on evenly-spaced times $\mathbf{u}$

$$f(t) = \frac{\sum_{i=1}^{L} K(u_i, t) v_i}{\sum_{i=1}^{L} K(u_i, t)}$$ *kernel smoother*



Continuous Convolutional Layer

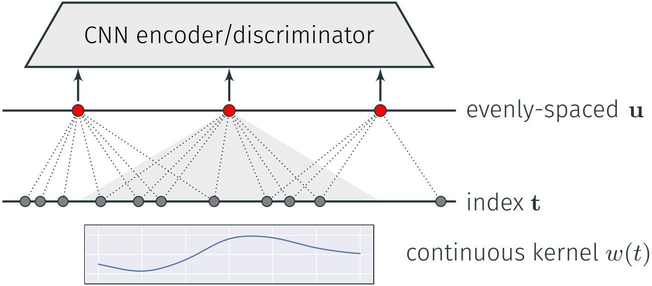


Cross-correlation between:

- continuous kernel $w(t)$
- masked function $m(\mathbf{x}, \mathbf{t})(t) = \sum_{i=1}^{|\mathbf{t}|} x_i \delta(t - t_i)$

$\delta(\cdot)$ is the Dirac delta function.

Continuous Convolutional Layer

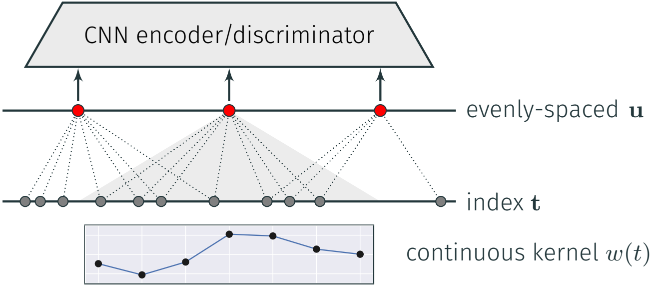


Cross-correlation between w and $m(\mathbf{x}, \mathbf{t})$:

$$(w \star m(\mathbf{x}, \mathbf{t}))(u) = \sum_{i : t_i \in \text{neighbor}(u)} w(t_i - u) x_i$$

Construct kernel $w(t)$ using a *degree-1 B-spline*

Continuous Convolutional Layer

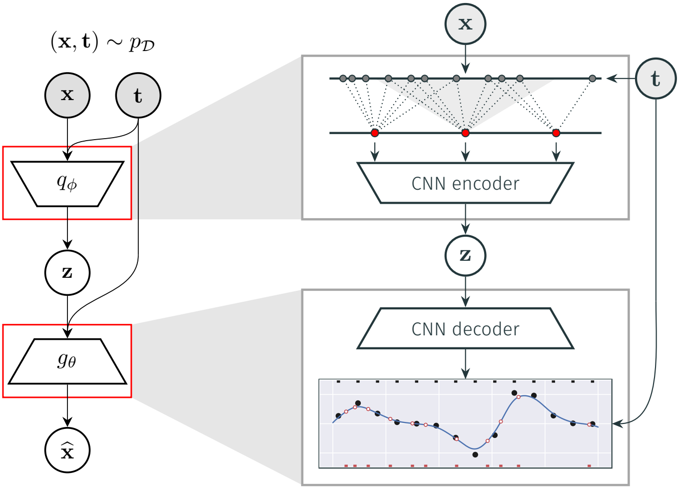


Cross-correlation between w and $m(\mathbf{x}, \mathbf{t})$:

$$(w \star m(\mathbf{x}, \mathbf{t}))(u) = \sum_{i:t_i \in \text{neighbor}(u)} w(t_i - u) x_i$$

Construct kernel $w(t)$ using a *degree-1 B-spline*

Architecture Overview for Continuous Time Series

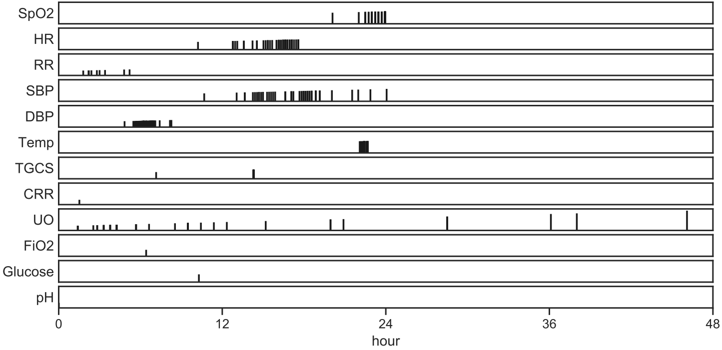


generic P-VAE

P-VAE for continuous time series

MIMIC-III Mortality Prediction

- about 53,000 labeled examples
- 12 irregularly-sampled physiological time series
- average mortality rate: 8.10%



MIMIC-III Mortality Prediction

method	AUC (%)	time (hr)	params
GRU-D†	83.88 ± 0.65	0.11	2.6K
Latent ODE‡	85.71 ± 0.38	2.62	154.7K
Cont classifier	84.87 ± 0.18	0.03	30.5K
Cont P-VAE	85.13 ± 0.43	0.04	64.8K
Cont P-BiGAN	$\mathbf{86.02 \pm 0.38}$	0.22	73.2K

†Che, et al. (2018). RNNs for multivariate time series with missing values.
‡Rubanova, et al. (2019). Latent ODEs for irregularly-sampled time series.

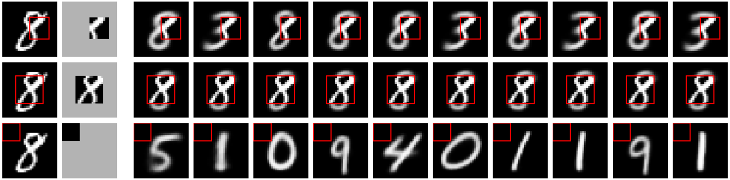
Summary

- Transforming modeling of irregularly-sampled time series into missing data problem
- An encoder-decoder framework for missing data problem
    - Partial VAE
    - Partial BiGAN
- Scalable architecture for modeling continuous time series
    - Kernel smoothing decoder
    - Continuous convolutional layer

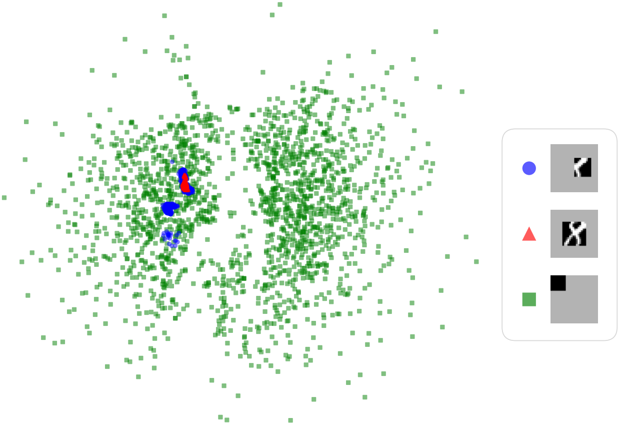
Appendix

Why Stochastic Encoders?

Imputation by model trained with 2-D latent code



Why Stochastic Encoders?



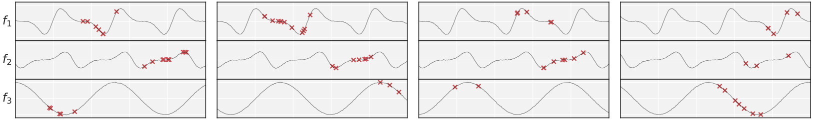
Different incomplete cases carry different levels of uncertainty

Synthetic Multivariate Time Series

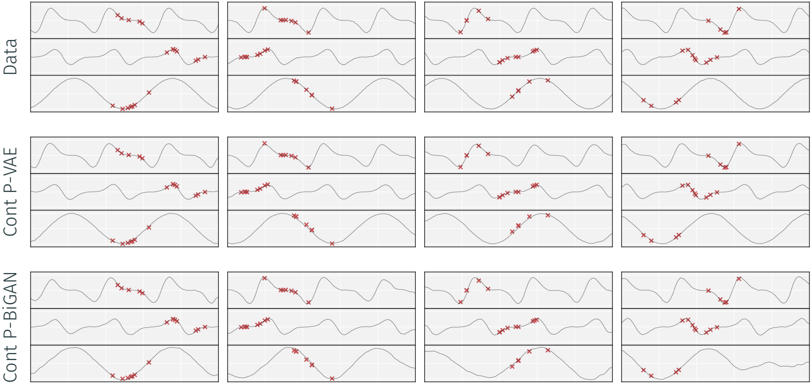
Generative process:

$$
\begin{aligned}
a &\sim \mathcal{N}(0, 10^2) \\
b &\sim \text{uniform}(0, 10) \\
f_1(t) &= 0.8 \sin(20(t+a) + \sin(20(t+a))) \\
f_2(t) &= -0.5 \sin(20(t+a+20) + \sin(20(t+a+20))) \\
f_3(t) &= \sin(12(t+b))
\end{aligned}
$$

Observation time points drawn from homogeneous Poisson process with $\lambda = 30$ within $[d, d+0.25]$ where $d \sim$ uniform$(0, 0.75)$.

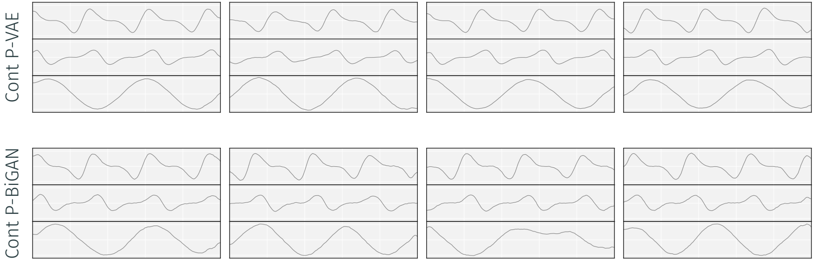


Synthetic Multivariate Time Series

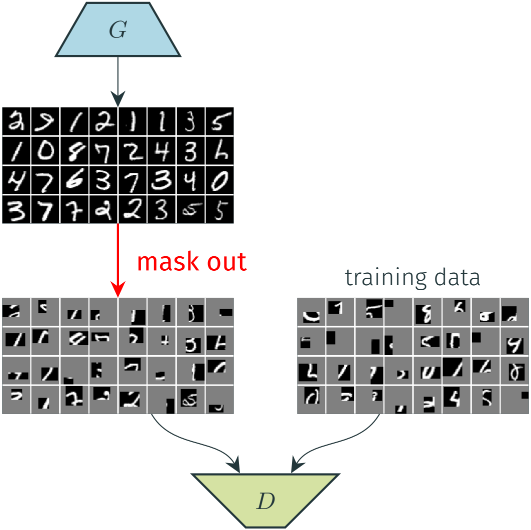


Synthetic Multivariate Time Series

Random time series generation:



MisGAN: GAN for Missing Data



Li, Jiang, Marlin. (2019). MisGAN: Learning from Incomplete Data with GANs.

On the Independence Assumption

For the most general case *without* the independence assumption, we use the generative process for an incomplete case $(\mathbf{x}, \mathbf{t})$:

$$\begin{aligned} \mathbf{z} &\sim p_z(\mathbf{z}) \\ \mathbf{t} &\sim p_{\mathcal{I}}(\mathbf{t}|\mathbf{z}) \\ \mathbf{x} &= g_\theta(\mathbf{z}, \mathbf{t}) \end{aligned}$$

It encodes dependency between $\mathbf{t}$ and $\mathbf{x}$ when $\mathbf{z}$ is unobserved.

On the Independence Assumption

Generative process for an incomplete case $(\mathbf{x}, \mathbf{t})$:

$$\mathbf{z} \sim p_z(\mathbf{z}), \quad \mathbf{t} \sim p_{\mathcal{I}}(\mathbf{t}|\mathbf{z}), \quad \mathbf{x} = g_\theta(\mathbf{z}, \mathbf{t}).$$

P-VAE:

$$\max_{\phi,\theta,\tau} \mathbb{E}_{(\mathbf{x},\mathbf{t})\sim p_{\mathcal{D}}} \mathbb{E}_{q_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})} \left[\log \frac{p_z(\mathbf{z}) p_{\mathcal{I}}(\mathbf{t}|\mathbf{z}) \prod_{i=1}^{|\mathbf{t}|} p_\theta(x_i|\mathbf{z}, t_i)}{q_\phi(\mathbf{z}|\mathbf{x}, \mathbf{t})} \right]$$

P-BiGAN:

$$\min_{\theta,\phi,\tau} \max_{D} \Big(\mathbb{E}_{(\mathbf{x},\mathbf{t})\sim p_{\mathcal{D}}} \mathbb{E}_{\mathbf{z}\sim p_\phi(\mathbf{z}|\mathbf{x},\mathbf{t})} \left[\log D(\mathbf{x}, \mathbf{t}, \mathbf{z})\right]$$

$$+ \mathbb{E}_{\mathbf{z}\sim p_z(\mathbf{z})} \mathbb{E}_{\mathbf{t}\sim p_{\mathcal{I}}(\mathbf{t}|\mathbf{z})} \left[\log(1 - D(g_\theta(\mathbf{z}, \mathbf{t}), \mathbf{t}, \mathbf{z}))\right] \Big)$$

τ denotes the parameters of $p_{\mathcal{I}}(\mathbf{t}|\mathbf{z})$.
For P-BiGAN, $p_{\mathcal{I}}(\mathbf{t}|\mathbf{z})$ can be stochastic or deterministic.