Guided Tour of Machine Learning in Finance

Week 4: Reinforcement Learning

4-1-2-Sequence modeling

Igor Halperin

NYU Tandon School of Engineering, 2017

Sequential data vs i.i.d. data

Factor analysis decomposes the observable N-dimensional signal $y = \left\{ y_i^{(t)} \right\}_{t=1}^{T}$ (where $i = 1,\ldots,N$) as a weighted sum of hidden (latent) uncorrelated Gaussian variables x with zero means and unit variances, plus a N-dimensional white noise ε with a diagonal correlation matrix Ψ:

$$y^{(t)} = \Lambda x^{(t)} + \varepsilon$$

Here x is a K-dimensional vector, and Λ is a factor loading matrix of size $N \times K$.
The i.i.d. data (a standard assumption in most of ML tasks!): $(x^{(t)}, y^{(t)}) \sim p_{data}(x, y)$

Sequential data vs i.i.d. data

Factor analysis decomposes the observable N-dimensional signal $y = \left\{ y_i^{(t)} \right\}_{t=1}^{T}$ (where $i = 1,\ldots,N$) as a weighted sum of hidden (latent) uncorrelated Gaussian variables x with zero means and unit variances, plus a N-dimensional white noise ε with a diagonal correlation matrix Ψ:

$$y^{(t)} = \Lambda x^{(t)} + \varepsilon$$

Here x is a K-dimensional vector, and Λ is a factor loading matrix of size $N \times K$.
The i.i.d. data (a standard assumption in most of ML tasks!): $(x^{(t)}, y^{(t)}) \sim p_{data}(x, y)$

But **most of data in Finance is sequential**!
- Asset prices (stocks, bonds, commodities, FX, etc.)
- Macro-economic data
- Balance sheet data
- News

Sequential data vs i.i.d. data

Factor analysis decomposes the observable N-dimensional signal $y = \left\{ y_i^{(t)} \right\}_{t=1}^{T}$ (where $i = 1,\ldots,N$) as a weighted sum of hidden (latent) uncorrelated Gaussian variables x with zero means and unit variances, plus a N-dimensional white noise ε with a diagonal correlation matrix Ψ:

$$y^{(t)} = \Lambda x^{(t)} + \varepsilon$$

Here x is a K-dimensional vector, and Λ is a factor loading matrix of size $N \times K$.
The i.i.d. data (a standard assumption in most of ML tasks!): $(x^{(t)}, y^{(t)}) \sim p_{data}(x, y)$

But **most of data in Finance is sequential**!
- Asset prices (stocks, bonds, commodities, FX, etc.)
- Macro-economic data
- Balance sheet data
- News

The i.i.d. assumption does **not** hold for sequential data!
i.i.d.: $(E_1(T_1), E_2(T_2)) \sim p_{data}(x,y)$ and $(E_1(T_1), E_2(T_2)) \sim p_{data}(x,y)$ have the same probabilities
- E_1- banking fraud in Labor manipulation, E_2- regulatory fines
- $(E_1(T_1), E_2(T_2)) \sim p_{data}(x,y)$ - first fraud, then fines
- $(E_1(T_1), E_2(T_2)) \sim p_{data}(x,y)$ - first fines, then fraud

Sequential data vs i.i.d. data

Factor analysis decomposes the observable N-dimensional signal $y = \left\{ y_i^{(t)} \right\}_{t=1}^{T}$ (where $i = 1,\ldots,N$) as a weighted sum of hidden (latent) uncorrelated Gaussian variables x with zero means and unit variances, plus a N-dimensional white noise ε with a diagonal correlation matrix Ψ

$$y^{(t)} = \Lambda x^{(t)} + \varepsilon$$

Here x is a K-dimensional vector, and Λ is a factor loading matrix of size $N \times K$.
The i.i.d. data (a standard assumption in most of ML tasks!): $(x^{(t)}, y^{(t)}) \sim p_{data}(x, y)$

But **most of data in Finance is sequential**!
- Asset prices (stocks, bonds, commodities, FX, etc.)
- Macro-economic data
- Balance sheet data
- News

The i.i.d. assumption does **not** hold for sequential data!
Taking differences (as e.g. for computing stock returns) is a poor man solution in general:
- Dependencies may persist beyond first differences
- For some discrete sequences, a differencing operation may be undefined.

How to model sequential data

The last: build a probabilistic model for the observable signal $y = \left\{ y^{(t)} \right\}_{t=1}^{T}$

$$p\left(y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(T)}\right) = \prod_{n=1}^{T} p\left(y^{(n)} \mid y^{(1)}, y^{(2)}, \ldots, y^{(n-1)}\right)$$

Possible modeling approaches for $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right)$ (need to constrain complexity of the model = keep the number of predictors low!)

How to model sequential data

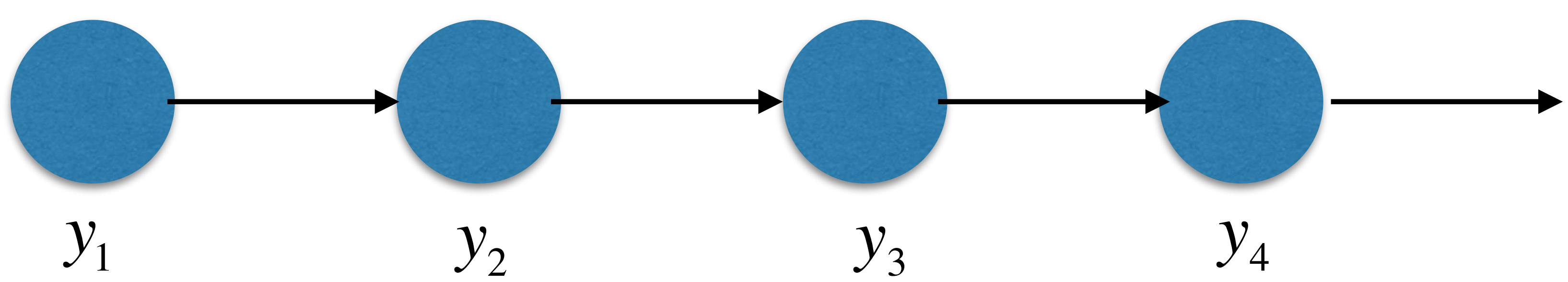
The last: build a probabilistic model for the observable signal $y = \left\{ y^{(t)} \right\}_{t=1}^{T}$

$$p\left(y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(T)}\right) = \prod_{n=1}^{T} p\left(y^{(n)} \mid y^{(1)}, y^{(2)}, \ldots, y^{(n-1)}\right)$$

Possible modeling approaches for $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right)$ (need to constrain complexity of the model = keep the number of predictors low!)

1. **Markov models:**

- First-order Markov: $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right) = p\left(y^{(t)} \mid y^{(t-1)}\right)$



How to model sequential data

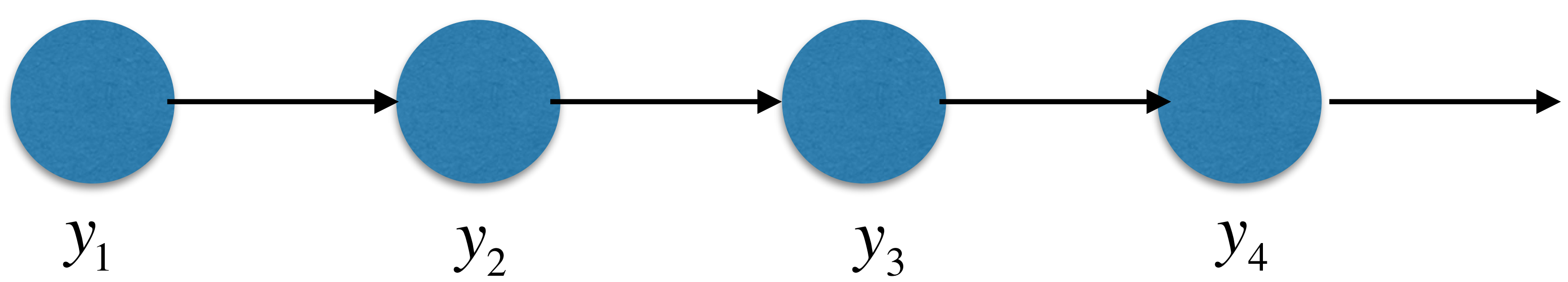
The last: build a probabilistic model for the observable signal $y = \left\{ y^{(t)} \right\}_{t=1}^{T}$

$$p\left(y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(T)}\right) = \prod_{n=1}^{T} p\left(y^{(n)} \mid y^{(1)}, y^{(2)}, \ldots, y^{(n-1)}\right)$$

Possible modeling approaches for $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right)$ (need to constrain complexity of the model = keep the number of predictors low!)

1. **Markov models:**

- First-order Markov: $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right) = p\left(y^{(t)} \mid y^{(t-1)}\right)$
- K-order Markov: $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right) = p\left(y^{(t)} \mid y^{(t-1)}, \ldots, y^{(t-K)}\right),\ K \geq 2$



How to model sequential data

The last: build a probabilistic model for the observable signal $y = \left\{ y^{(t)} \right\}_{t=1}^{T}$

$$p\left(y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(T)}\right) = \prod_{n=1}^{T} p\left(y^{(n)} \mid y^{(1)}, y^{(2)}, \dots, y^{(n-1)}\right)$$

Possible modeling approaches for $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \dots\right)$ (need to constrain complexity of the model = keep the number of predictors low!)

1. **Markov models:**

- First-order Markov: $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \dots\right) = p\left(y^{(t)} \mid y^{(t-1)}\right)$
- K-order Markov: $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \dots\right) = p\left(y^{(t)} \mid y^{(t-1)}, \dots, y^{(t-K)}\right),\ K \geq 2$
- Potential issues with Markov model:
  - Sometimes would need a large K to capture history -> but would lead to a parameter proliferation.
  - As the signal $y^{(t)}$ itself is noisy, conditioning on its past values in may not be optimal

Control question

Select all correct answers

1. Sequential models are models that are written one after another by the same ML researcher, after the previous model does not perform well.
2. A First-Order Markov model is obtained from a conventional Markov model when its predictions are first ordered in values from smallest to the largest ones.
3. A K-order Markov model is obtained by running K first-order Markov models in parallel.
4. The next value of a Markov model depends only on the previous value, therefore for stock prices, it implies that the next return depends only on the previous one.
5. First-order Markov models are memoryless, which means that the next value is totally random, and is independent of the past history.

Correct answers: none

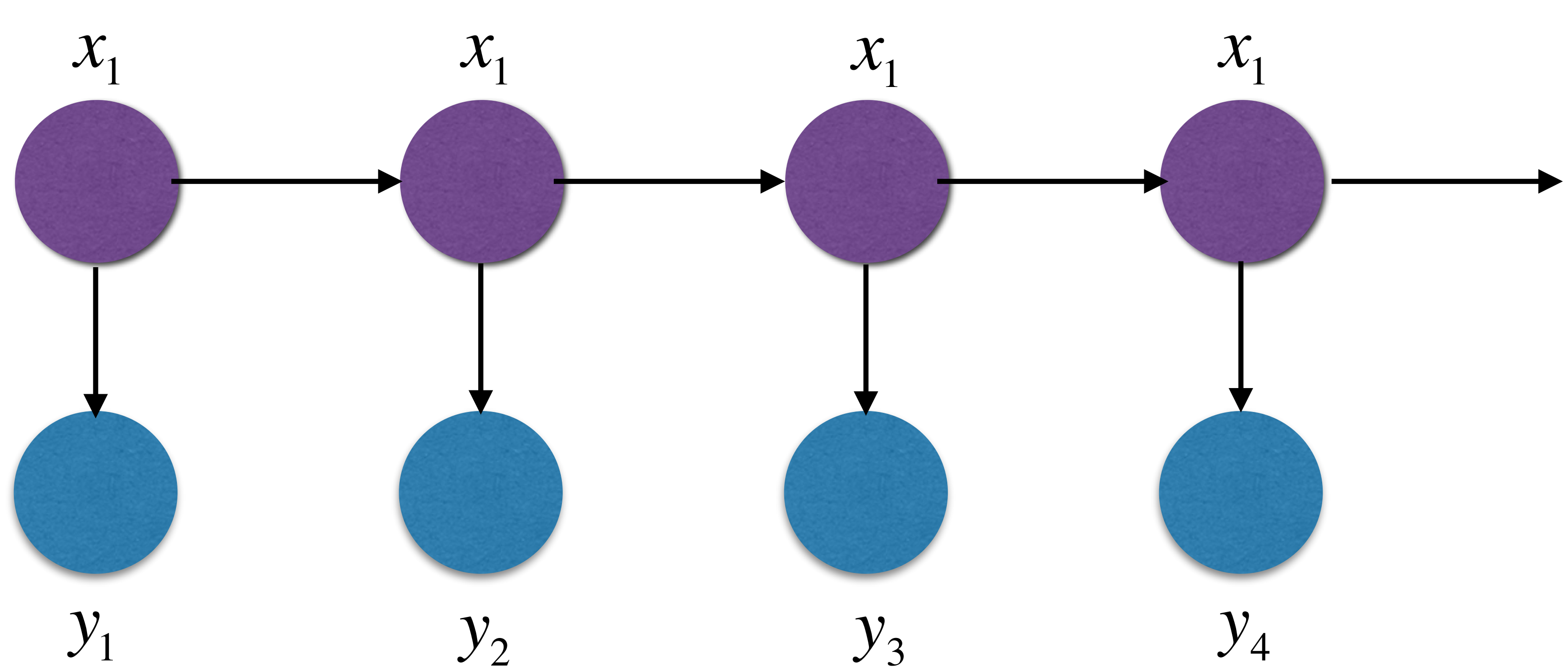
How to model sequential data

The last: build a probabilistic model for the observable signal $y = \left\{ y^{(t)} \right\}_{t=1}^{T}$

$$p\left(y^{(1)}, y^{(2)}, y^{(3)}, \ldots, y^{(T)}\right) = \prod_{n=1}^{T} p\left(y^{(n)} \mid y^{(1)}, y^{(2)}, \ldots, y^{(n-1)}\right)$$

Possible modeling approaches for $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \ldots\right)$:

1. **Markov models:**
2. **Dynamic hidden (latent) variables models**, with an unobservable state



How to model sequential data

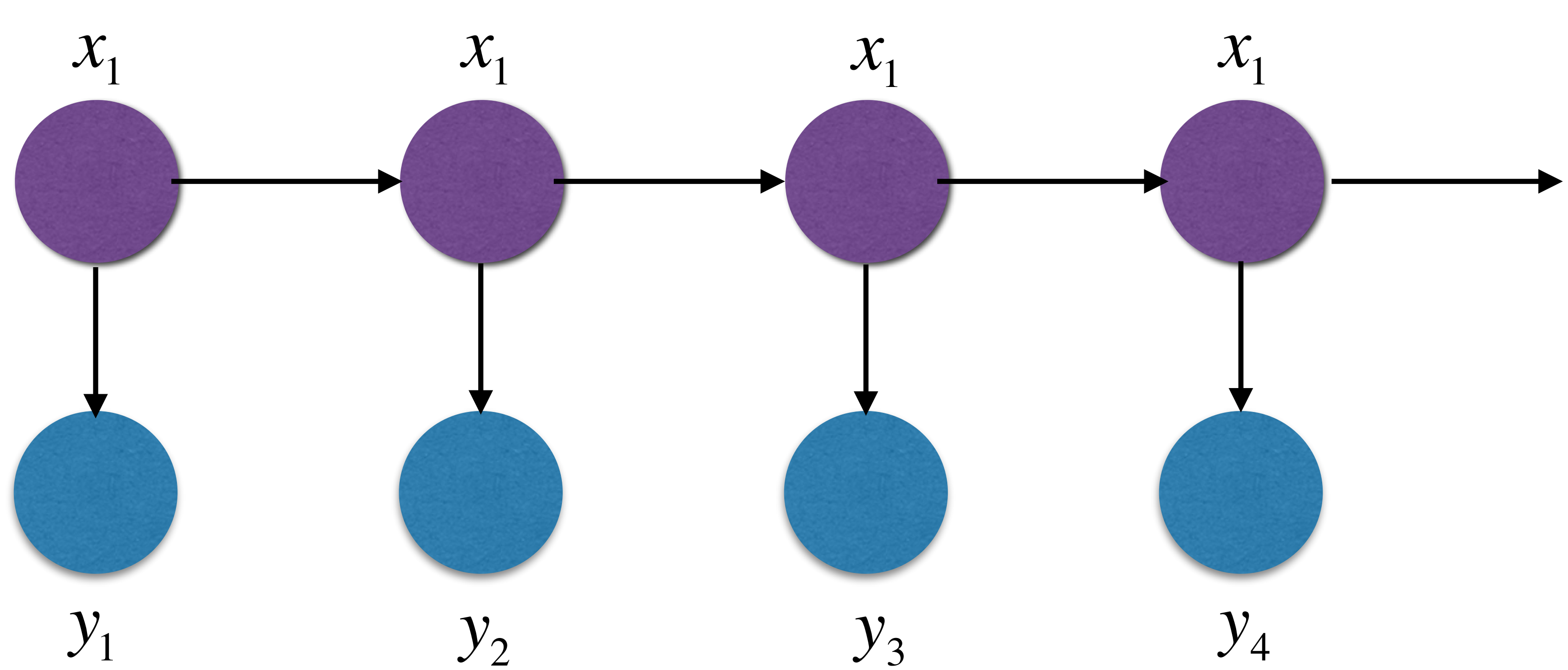
The last: build a probabilistic model for the observable signal $y = \left\{ y^{(t)} \right\}_{t=1}^{T}$

$$p\left(y^{(1)}, y^{(2)}, y^{(3)}, \dots, y^{(T)}\right) = \prod_{n=1}^{T} p\left(y^{(n)} \mid y^{(1)}, y^{(2)}, \dots, y^{(n-1)}\right)$$

Possible modeling approaches for $p\left(y^{(t)} \mid y^{(t-1)}, y^{(t-2)}, \dots\right)$:

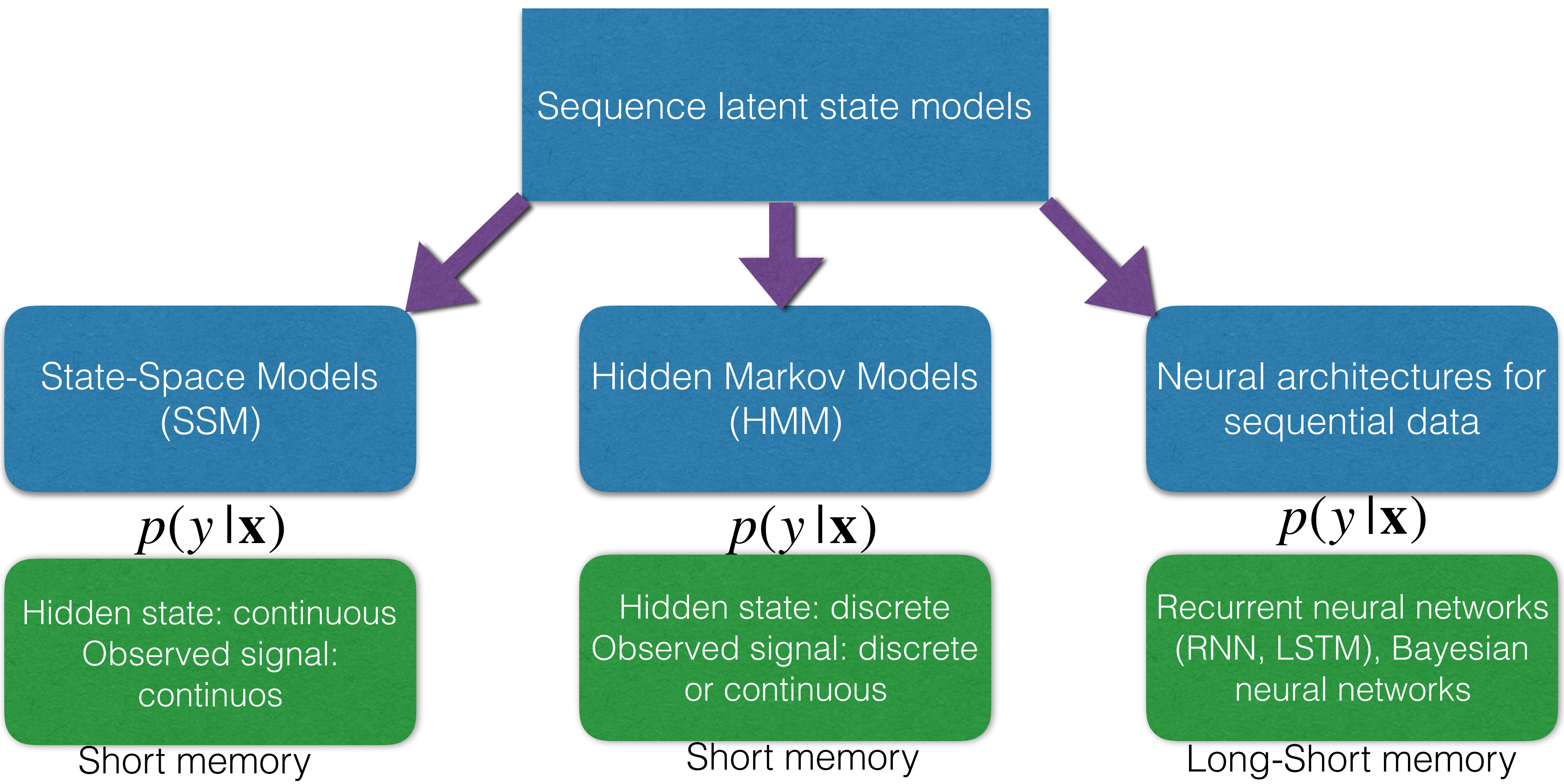
1. **Markov models:**
2. **Dynamic hidden (latent) variables models**, with an unobservable state $x^{(t)}$

- The hidden state $x^{(t)}$ captures the dynamics of the system, filters noise out
- Used as a conditioning variable for predictions $p\left(y^{(t)} \mid y^{(t-1)}\right) \rightarrow p\left(y^{(t)} \mid x^{(t)}\right)$.



Sequence latent state models

Parametric (SSM, HMM) vs non-parametric (neural) of sequence modeling with a hidden state $p(y\,|\,\mathbf{x})$



Control question

Select all correct answers

1. State-Space models measure the fraction of total space occupied a given state of a system, then assign probabilities to all states proportional to these fractions.
2. The Hidden Markov Models (HMM) are named so because in these models the Markov dynamics is hidden (masked by observational noise), therefore they model the dynamics as non-Markov.
3. For State-Space Models (SSM), both the hidden and observed states are continuous.
4. For Hidden Markov Models (HMM), the hidden state is discrete, while the observed state can be either discrete or continuous
5. For Neural models of sequential data, both the hidden and observed states can be either discrete or continuous, as long as they are defined as TensorFlow constants on the graph.

Correct answers: 3, 4