Assorted Topics (2)

CS772A: Probabilistic Machine Learning

Piyush Rai

Plan Today

- Some other non-Bayesian ways to get uncertainty estimates
  - Frequentist statistics
  - Conformal Prediction
- Latent variable models for sequential data

Frequentist Statistics

- The Bayesian approach treats parameters/model unknowns as random variables
- In the Bayesian approach, the posterior over these r.v.'s help capture the uncertainty
- The Frequentist approach is a different way to capture uncertainty
  - Don't treat parameters as r.v. but as fixed unknowns
  - Treat parameters as a function of the dataset, e.g., $\hat{\theta}(\mathcal{D}) = \pi(\mathcal{D})$

This can be some point estimate, e.g., MLE, MAP, method of moments, etc.

  - Variations in param estimates over different datasets represents their uncertainty

A random dataset drawn from the true data distribution

True unknown value of the parameter

$$\tilde{\mathcal{D}}^{(s)} = \{\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\boldsymbol{\theta}^*) : n = 1 : N\} \qquad (s = 1, 2, \ldots, S)$$

The estimated distribution of the parameters given any randomly drawn dataset from the true data distribution

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S} \delta(\boldsymbol{\theta} = \pi(\tilde{\mathcal{D}}^{(s)}))$$

Param estimate using the s-th sampled dataset

As $S \to \infty$, this is known as the "sampling distribution" of the estimator

Note that sampling distribution is different from a posterior distribution we infer in Bayesian learning (there, we condition on a fixed training set)

But if the estimator is MLE and Bayesian method's prior is uniform, then both distributions are very similar (sampling distribution is often called "poor man's posterior"

Approximating the sampling distribution

- Since the true $\boldsymbol{\theta}^*$ is not known, we can't compute the sampling distribution exactly

$$\tilde{\mathcal{D}}^{(s)} = \{\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\boldsymbol{\theta}^*) : n = 1 : N\} \qquad (s = 1, 2, \ldots, S)$$

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S} \delta(\boldsymbol{\theta} = \pi(\widetilde{\mathcal{D}^{(s)}}))$$

- Bootstrap is a popular method to approximate the sampling distribution
- Two types of bootstrap methods: parametric and nonparametric bootstrap

Parametric Bootstrap

- Get a point est. of $\boldsymbol{\theta}$ using training data

$$\hat{\boldsymbol{\theta}} = \pi(\mathcal{D})$$

- Generate multiple datasets using $\hat{\boldsymbol{\theta}}$ as

$$\tilde{\mathcal{D}}^{(s)} = \{\boldsymbol{x}_n \sim p(\boldsymbol{x}_n|\hat{\boldsymbol{\theta}}) : n = 1 : N\} \quad (s = 1, 2, \ldots, S)$$

- Now compute the approximation as

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S} \delta(\boldsymbol{\theta} = \pi(\widetilde{\mathcal{D}^{(s)}}))$$

Nonparametric Bootstrap

- Use sampling with replacement on original training set to generate S datasets with N datapoints in each

Each dataset will contain roughly 63% unique datapoints from original training set

- Now compute the approximation as

$$p(\pi(\tilde{\mathcal{D}}) = \boldsymbol{\theta}|\tilde{\mathcal{D}} \sim \boldsymbol{\theta}^*) \approx \frac{1}{S}\sum_{s=1}^{S} \delta(\boldsymbol{\theta} = \pi(\widetilde{\mathcal{D}^{(s)}}))$$

Conformal Prediction

Conformal Prediction

- A simple technique to easily obtain confidence intervals
  - In classification, such an interval may refer to the set of highly likely classes for a test input

  - For more difficult test inputs, the set would typically be larger
- In a way, conformal prediction gives predictive uncertainty
  - However, unlike Bayesian ML, we don't get model uncertainty
  - Only one model is learned in the standard way and we construct the set of likely classes
  - It's like a black-box method; no change to training procedure for the model

Conformal Prediction

Assume it's a classification model which produces softmax scores

Conformal prediction can be used for regression problems too*

- Assume we already have a trained model $\hat{f}$ using some labelled data
- Suppose we get a test input X_{test} whose true (unknown) label is Y_{test}
- Use $\hat{f}$ and a calibration set of n examples to generate a prediction set $\mathcal{C}(X_{test})$ s.t.

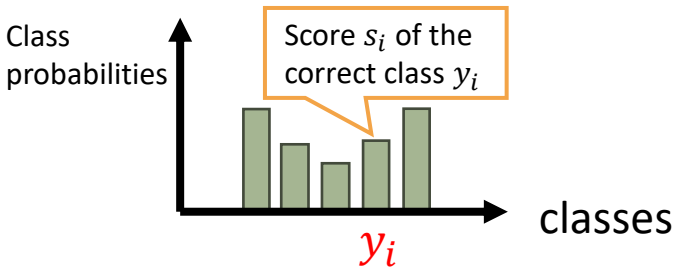
α is a user chosen error rate

$$1 - \alpha \leq p\left(Y_{test} \in \mathcal{C}(X_{test})\right) \leq 1 - \alpha + \frac{1}{n+1}$$

- To construct the set, we first compute, for each example in the calibration set

Probability/score of the correct class y_i of the input x_i

$$s_i = \hat{f}(x_i)_{y_i}$$

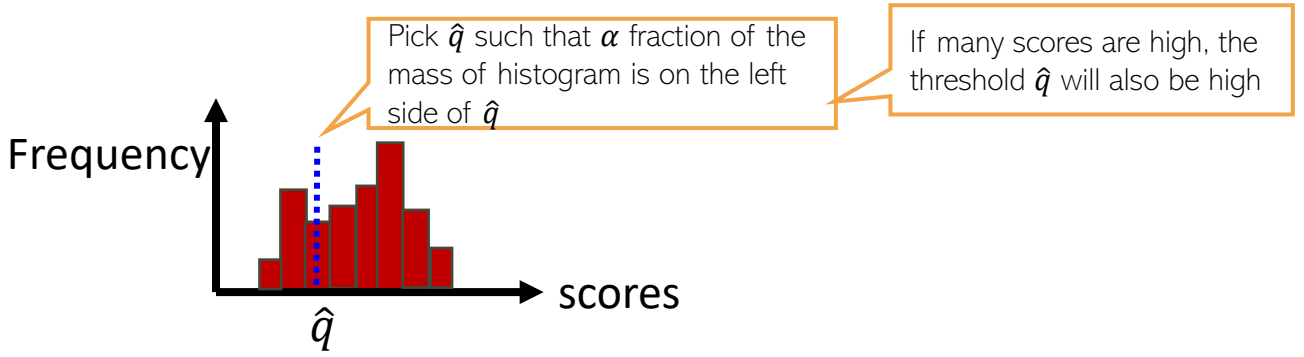


- Use the calibration set scores $s_1, s_2, \ldots, s_n$ to compute their α quantile

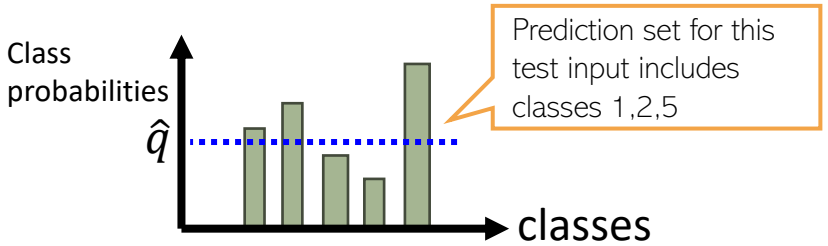
*A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification (Angelopoulos and Bates, 2022)

Conformal Prediction

- Assume the α (say 0.1) quantile of the calibration set scores is equal to $\hat{q}$ ("threshold")



- Given a test input X_{test}, whose label is is unknown, we compute the class probabilities



- Define the prediction set for X_{test} as: $\mathcal{C}(X_{test}) = \{y: \hat{f}(X_{test})_y \geq \hat{q}\}$

Report all the classes whose probability is large enough (the "large enough" value is given by the threshold $\hat{q}$)

- The above ensures that

$$1 - \alpha \leq p\left(Y_{test} \in \mathcal{C}(X_{test})\right) \leq 1 - \alpha + \frac{1}{n+1}$$

For large n, α fraction of test inputs will have their true label not in the predicted set

Conformal Prediction

- A generic black-box method
- Can be easily applied to any already trained classifier
- Predicted set has some nice guarantees

$$1 - \alpha \leq p\left(Y_{test} \in \mathcal{C}(X_{test})\right) \leq 1 - \alpha + \frac{1}{n+1}$$

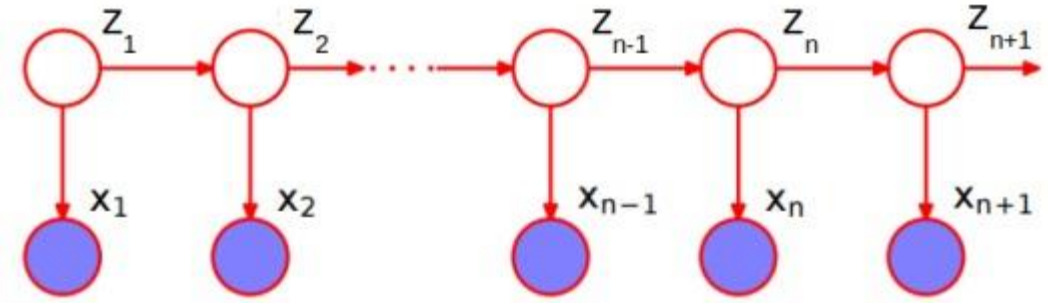
- Does not make any assumptions on the distribution of the data
  - Thus considered a "distribution-free" approach to uncertainty quantification
- Can also be applied to regression problems*

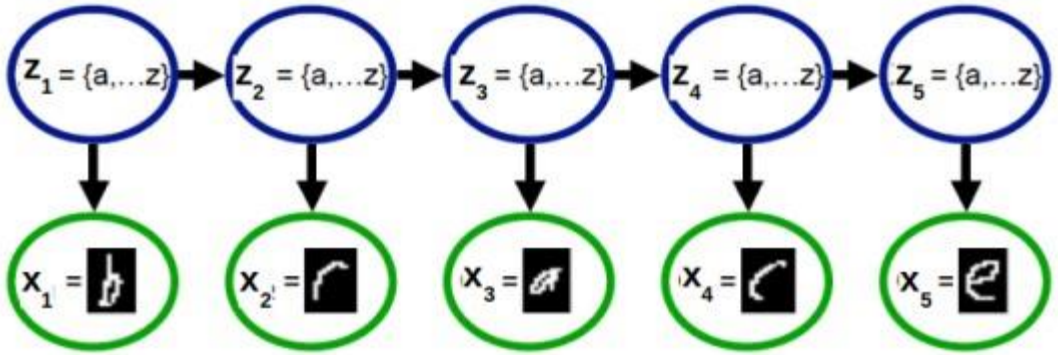
*A Gentle Introduction to Conformal Prediction and Distribution-Free Uncertainty Quantification (Angelopoulos and Bates, 2022)

Latent Variable Models for Sequential Data

Latent Variable Models for Sequential Data

- Task: Given a sequence of observations, infer the latent state of each observation





Observation model

State-transition model

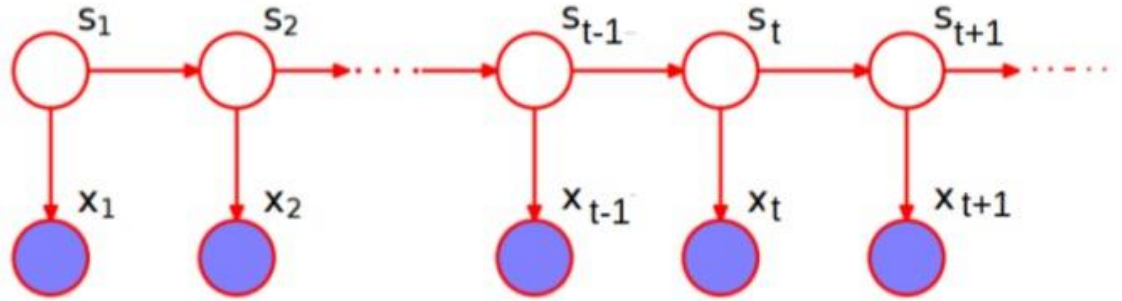
$$\boldsymbol{x}_n|\boldsymbol{z}_n \sim p(\boldsymbol{x}_n|\boldsymbol{z}_n) \quad \text{(i.i.d. draws of } \boldsymbol{x}_n \text{ given } \boldsymbol{z}_n\text{)}$$

$$\boldsymbol{z}_n|\boldsymbol{z}_{n-1} \sim p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) \quad \text{(first-order dependence b/w } \boldsymbol{z}_n\text{'s)}$$

- If $\boldsymbol{z_n}$'s are discrete, we have a hidden Markov model (HMM) $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1} = \ell) = \text{multinoulli}(\boldsymbol{\pi}_\ell)$
- If $\boldsymbol{z_n}$'s are real-valued, we have a state-space model (SSM) $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) = \mathcal{N}(\mathbf{A}\boldsymbol{z}_{n-1}, \mathbf{I}_K)$

State-Space Models

- In the most general form, the state-transition and observation models of an SSM



Using '**s**' instead of '**z**' to refer to states

Using 't' to denote the 'time-step'

HMM is similar to SSM except the state-transition model is a discrete distribution

g_t, h_t can be linear or nonlinear functions

$$\boldsymbol{s}_t|\boldsymbol{s}_{t-1} = g_t(\boldsymbol{s}_{t-1}) + \epsilon_t \quad \text{(must be a cont. dist. over } \boldsymbol{s}_t\text{)}$$

$$\boldsymbol{x}_t|\boldsymbol{s}_t = h_t(\boldsymbol{s}_t) + \delta_t \quad \text{(can be any dist. over } \boldsymbol{x}_t\text{)}$$

- Assuming Gaussian noise in the state-transition and observation models

This is a Gaussian SSM

$$\boldsymbol{s}_t|\boldsymbol{s}_{t-1} \sim \mathcal{N}(\boldsymbol{s}_t|g_t(\boldsymbol{s}_{t-1}), \mathbf{Q}_t)$$

$$\boldsymbol{x}_t|\boldsymbol{s}_t \sim \mathcal{N}(\boldsymbol{x}_t|h_t(\boldsymbol{s}_t), \mathbf{R}_t)$$

If g_t, h_t, Q_t, R_t are independent of t then it is called a stationary model

g_t, h_t, Q_t, R_t may be known or can be learned

State-Space Models: A Simple Example

- Consider the linear Gaussian SSM

$$\boldsymbol{s}_t|\boldsymbol{s}_{t-1} = \mathbf{A}_t\boldsymbol{s}_{t-1} + \epsilon_t$$

$$\boldsymbol{x}_t|\boldsymbol{s}_t = \mathbf{B}_t\boldsymbol{s}_t + \delta_t$$

- Suppose $\boldsymbol{x}_t \in \mathbb{R}^2$ denotes the (noisy) observed 2D location of an object
- Suppose $\boldsymbol{s}_t \in \mathbb{R}^6$ denotes the "state" vector

$\boldsymbol{s}_t$ = [pos1, vel1, accel1, pos2, vel2, accel2]

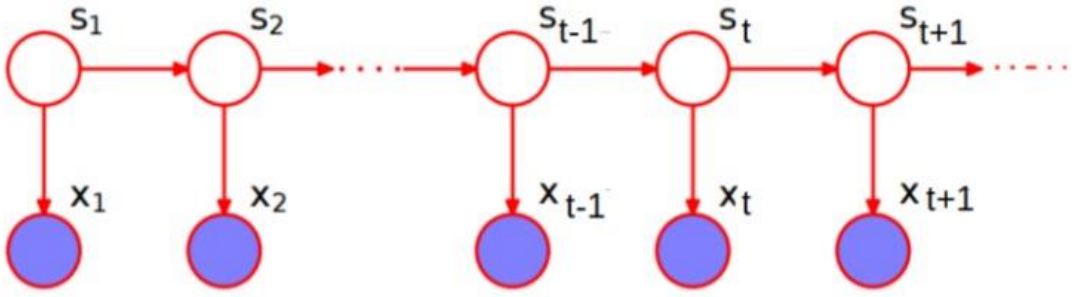
- Here is an example SSM for this problem with pre-defined $\mathbf{A_t}$ and $\mathbf{B_t}$ matrices

$$\mathbf{s}_t = \underbrace{\begin{bmatrix} 1 & \Delta t & \frac{1}{2}(\Delta t)^2 & 0 & 0 & 0 \\ 0 & 1 & \Delta t & 0 & 0 & 0 \\ 0 & 0 & e^{-\alpha\Delta t} & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & \Delta t & \frac{1}{2}(\Delta t)^2 \\ 0 & 0 & 0 & 0 & 1 & \Delta t \\ 0 & 0 & 0 & 0 & 0 & e^{-\alpha\Delta t} \end{bmatrix}}_{\mathbf{A}_t} \mathbf{s}_{t-1} + \epsilon_t$$

$$\mathbf{x}_t = \underbrace{\begin{bmatrix} 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 \end{bmatrix}}_{\mathbf{B}_t} \mathbf{s}_t + \delta_t$$

Typical Inference Task for Gaussian SSM

- One of the key tasks: Given sequence $x_1, x_2, \ldots, x_T$, infer latent $s_1, s_2, \ldots, s_T$



- Usually two ways of inferring the latent states
  - Infer $p(s_t|x_1, x_2, \ldots, x_t)$: Called the "filtering" problem

Turns out to be another Gaussian

$$p(s_t|x_1, x_2, \ldots, x_t) \propto \underbrace{p(x_t|s_t)}_{\mathcal{N}(x_t|Bs_t,R)} \int \underbrace{p(s_t|s_{t-1})}_{\mathcal{N}(s_t|As_{t-1},Q)} p(s_{t-1}|x_1, x_2, \ldots, x_{t-1}) ds_{t-1}$$

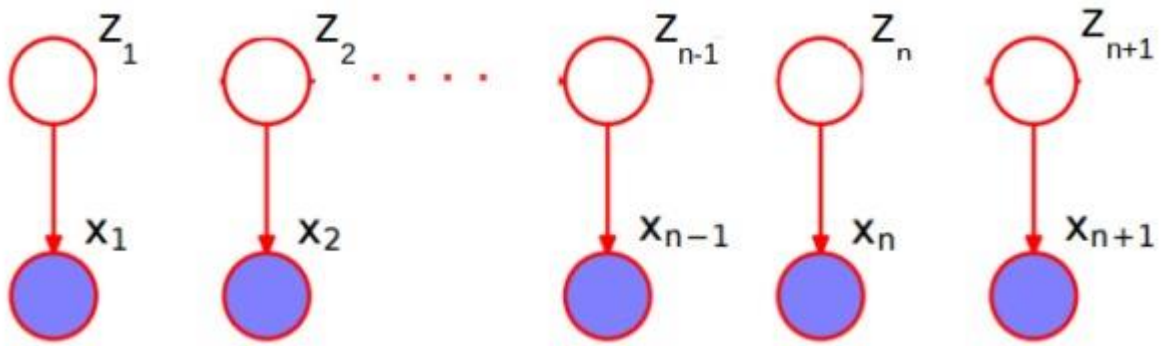
A Gaussian

Kalman Filtering is a popular algorithm for a linear Gaussian SSM

  - Infer $p(s_t|x_1, x_2, \ldots, x_t, \ldots, x_T)$: Called the "smoothing" problem
- Some other tasks one can solve for using an SSM
  - Predicting future states $p(s_{t+h}|x_1, x_2, \ldots, x_t)$ for $h \geq 1$, given observations thus far
  - Predicting future observations $p(x_{t+h}|x_1, x_2, \ldots, x_t)$ for $h \geq 1$, given observations thus far

A Special Case

- What if we have i.i.d. latent states, i.e.,. $p(z_n|z_{n-1}) = p(z_n)$?



- Discrete case (HMM) becomes a simple mixture model $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1} = \ell) = p(\boldsymbol{z}_n) = \text{multinoulli}(\boldsymbol{\pi})$
- Real-valued case (SSM) becomes a PPCA model $p(\boldsymbol{z}_n|\boldsymbol{z}_{n-1}) = p(\boldsymbol{z}_n) = \mathcal{N}(\mathbf{0}, \mathbf{I}_K)$ or $\mathcal{N}(\boldsymbol{\mu}, \Psi)$
- Inference algos for HMM/SSM are thus very similar to that of mixture models/PPCA
  - Only main difference is how the latent variables $\boldsymbol{z}_n$'s are inferred since they aren't i.i.d.
  - E.g., if using EM, only E step needs to change (Bishop Chap 13 has EM for HMM and SSM)