Probabilistic Numerics, and Conclusion

Piyush Rai

Topics in Probabilistic Modeling and Inference (CS698X)

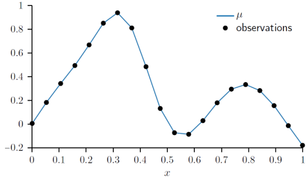
April 19, 2018

Classical Numerics

- An example problem: Imagine calculating the following definite integral

$$\int_0^1 \exp\left(-\frac{(x-0.35)^2}{2(0.1)^2}\right) + \frac{\sin(10x)}{3}\,\mathrm{d}x$$

- Suppose we don't know how to compute it. One option is numerical integration (quadrature)
- A classic approach to numerical integration: Trapezoid rule



$$\int_0^1 f(x)\,\mathrm{d}x \approx 0.3104$$

- Question: How many points to use? What's our <u>uncertainty</u> in $Z = \int_0^1 f(x)dx$? Where to measure $f(x)$ next to improve our estimate of Z? Calls for a probabilistic approach!

A Probabilistic Approach

- Let's treat our quantity of interest $Z = \int_0^1 f(x)dx$ as a random variable
- We can now assume a prior on Z and infer its posterior given some "observations"
- It is often easier to put a prior on f (even though f is "known") instead of on Z

$$p(f) = \mathcal{GP}(f|\mu, k)$$

- Reason: The integration operator is a linear functional and GPs are closed under linear functionals $L : f \mapsto L[f]$ (just like Gaussians under lin. trans.). Thus for $Z = L[f]$, the prior will be another GP

$$p(L[f]) = \mathcal{GP}(L[f]|L[\mu], L^2[k])$$

  where $L^2[k] = L[L[k]]$ (for a similar result, recall linear transformation of Gaussian r.v.'s)
- A functional $L[f]$ takes as input a function f and returns a scalar
- A functional $L[.]$ is a linear functional if it satisfies the linearity property, i.e.,

$$L[af + g] = aL[f] + L[g]$$

Examples of Linear Functionals

- Pointwise Evaluations: Values of f at any point x

$$L_x[f] = f(x)$$

- Definite Integrals: Definite integrals over arbitrary functions $p(x)$ (note: $p(x)$ can be a constant)

$$I_p[f] = \int_{\mathcal{X}} f(x)p(x)dx$$

- Partial Derivatives: Partial derivative of f at any point x

$$D_{x,i}[f] = \frac{\partial f(z)}{\partial z_i}\bigg|_{z=x}$$

A Probabilistic Approach (Contd.)

- We saw that a Gaussian Process is closed under linear functionals
  - If $p(f) = \mathcal{GP}(f|\mu, k)$ and we have a linear functional evaluation $\ell = L[f]$ of f, then the prior

$$p(\ell) = \mathcal{N}(\ell|L[\mu], L^2[k])$$

- Some examples of this property
  - Example 1: For the point-evaluation functional $L_x[f] = f(x)$, we will have the prior

$$p(f(x)) = \mathcal{N}(f(x)|L_x[\mu], L_x^2[k]) = \mathcal{N}(f(x)|\mu(x), k(x,x)),$$

  - Example 2: For the integration functional $L[f] = \int f(x)p(x)dx$, we will have the prior

$$p\left(\int f(x)p(x)dx\right) = \mathcal{N}\left(\int f(x)p(x)dx \middle| \int \mu(x)p(x)dx, \int\int k(x,x')p(x)p(x')dxdx'\right)$$

- Example 2 (computing definite integrals) is what we will look at today

Bayesian Quadrature for Definite Integrals

- As we saw, the integral has a GP prior (Gaussian in the finite case)

$$p\left(\int f(x)p(x)dx\right) = \mathcal{N}\left(\int f(x)p(x)dx \Big| \int \mu(x)p(x)dx, \int\int k(x,x')p(x)p(x')dxdx'\right)$$

- Suppose we are interested in the definite integral $Z = \int_0^1 f(x)dx$ (i.e., $p(x) = 1, \forall x$)
- The prior on this definite integral will be the following **Gaussian**

$$p\left(\int_0^1 f(x)dx\right) = \mathcal{N}\left(Z \Big| \int_0^1 \mu(x)dx, \int_0^1\int_0^1 k(x,x')dxdx'\right)$$

- Given observations $\boldsymbol{y}$ s.t. $y_n = f(\boldsymbol{x}_n) + \epsilon_n$, we can estimate the posterior $p(\int_0^1 f(x)dx|\boldsymbol{y})$
  - If $\epsilon_n = 0$ (or if it is zero-mean Gaussian noise), the posterior $p(\int_0^1 f(x)dx|\boldsymbol{y})$ will also be **Gaussian**

Bayesian Quadrature: An Example

- Suppose $f(x) = \exp[-\sin^2(3x) - x^2]$ and suppose we want to compute $Z = \int_{-3}^{3} f(x)dx$
- Assume the following zero mean GP prior on f, i.e.,

$$p(f) = \mathcal{GP}(f|0, k)$$

with $k(x, x') = c(1 + b - 1/3|x - x'|)$ for some $c, b > 0$

- Since GP is closed under linear functionals, the prior on $Z = \int_{-3}^{3} f(x)dx$ is a univariate Gaussian

$$p(Z) = \mathcal{N}\left[Z \middle| 0, \int_{-3}^{3}\int_{-3}^{3} k(x, x')dxdx' = c(1 + b/3)\right]$$

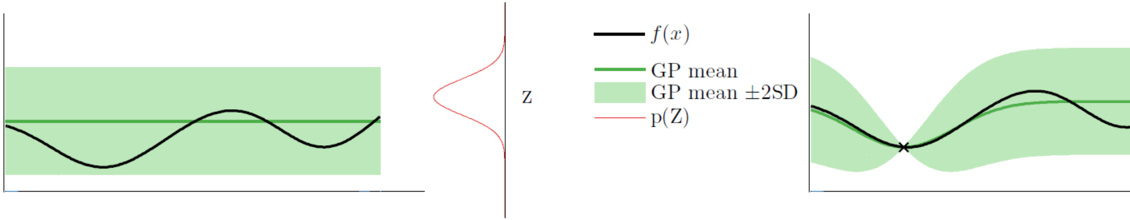
- Given observations $\boldsymbol{y}$ s.t. $y_n = f(\boldsymbol{x}_n) + \epsilon_n$, the posterior of Z will be

$$p(Z|\boldsymbol{y}) = \mathcal{N}\left[Z \middle| \int_{-3}^{3} k(x, \mathbf{X})\mathbf{K}^{-1}\boldsymbol{y}dx, \int_{-3}^{3}\int_{-3}^{3} k(x, x') - k(x, \mathbf{X})\mathbf{K}^{-1}k(\mathbf{X}, x')dxdx'\right]$$

where $\mathbf{K}$ is the kernel matrix over the $\mathbf{X} = [\boldsymbol{x}_1, \dots, \boldsymbol{x}_N]$ and $k(x, \mathbf{X}) = [k(\boldsymbol{x}, \boldsymbol{x}_1), \dots, k(\boldsymbol{x}, \boldsymbol{x}_N)]$

Bayesian Quadrature

- An illustration of the Bayesian Quadrature approach:



Bayesian Quadrature: Benefits and Limitations

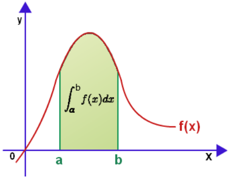
- Some of the benefits include
  - We can model the structure of f using the covariance function k
  - Thus Bayesian quadrature is also called "model based integration"
  - Tends to require far fewer samples than standard Monte-Carlo integral
  - Can design active sampling scheme to get most informative points (w.r.t. the precision of the integral being evaluated): Sequential Bayesian Quadrature
  - Naturally get the variance in our estimate of the integral
- Some of the limitations include
  - Works well usually in low-dimensions and if f is smooth
  - Usually worth it only when $f(x)$ is expensive to compute

Function Approximation using Functionals

- Let's consider another (standard) problem: Function approximation (i.e., regression)
- Assume a function $f : \mathcal{X} \to \mathbb{R}$ with a GP prior

$$p(f) = \mathcal{GP}(f|\mu, k)$$

- We know how to infer f using data of the form $(\boldsymbol{x}_n, y_n)_{n=1}^N$, assuming $y_n = f(\boldsymbol{x}_n) + \epsilon_n$
- The Twist: What if the training data to learn f is given in form of linear functionals of f? E.g.,
  - Values of definite integrals of f over some ranges, e.g., $\int_{a_1}^{b_1} f(x)dx, \ldots, \int_{a_N}^{b_N} f(x)dx$



  - Values of f's partial derivatives at some points

Function Approximation using Functionals

- Given a functional observation $\ell = L[f]$, how can we infer f, given a $\mathcal{GP}(f|\mu, K)$ prior on f?
- Suppose $\boldsymbol{f} = f(\mathbf{X})$ where $\mathbf{X} = \{\boldsymbol{x}_i\}$ denotes a set of arbitrary inputs
- The joint distribution of ℓ and $\boldsymbol{f}$ will be

$$p\left(\begin{bmatrix}\ell\\ \mathbf{f}\end{bmatrix} \mid \mathbf{X}\right) = \mathcal{N}\left(\begin{bmatrix}\ell\\ \mathbf{f}\end{bmatrix}; \begin{bmatrix}L[\mu]\\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}L^2[K] & ?\\ ? & \mathbf{K}\end{bmatrix}\right)$$

$$\boldsymbol{\mu} = \mu(\mathbf{X}) \qquad \mathbf{K} = K(\mathbf{X}, \mathbf{X})$$

where the "?" denotes the covariances of ℓ and $\boldsymbol{f} = \{f(\boldsymbol{x}_i)\}$, and are given by

$$\mathrm{cov}(\ell, f_i) = \mathrm{cov}\left(L[f], L_{\mathbf{x}_i}[f]\right) = L\left[L_{\mathbf{x}_i}\left[\mathrm{cov}(f, f)\right]\right] = L\left[L_{\mathbf{x}_i}\left[K\right]\right] = L\left[K(\cdot, \mathbf{x}_i)\right]$$

.. which follows from the linearity of covariance

Function Approximation using Functionals

- We will therefore have

$$p\left(\begin{bmatrix}\ell \\ \mathbf{f}\end{bmatrix} \mid \mathbf{X}\right) = \mathcal{N}\left(\begin{bmatrix}\ell \\ \mathbf{f}\end{bmatrix}; \begin{bmatrix}L[\mu] \\ \boldsymbol{\mu}\end{bmatrix}, \begin{bmatrix}L^2[K] & L\left[K(\cdot, \mathbf{X})\right] \\ L\left[K(\mathbf{X}, \cdot)\right] & \mathbf{K}\end{bmatrix}\right)$$

- Using the above, we can easily get $p(\boldsymbol{f}|\ell)$ using the Gaussian conditioning equations!
- The predictive posterior $p(f(\boldsymbol{x})|\ell)$ will be a Gaussian with following mean and variance

$$\mu_{f|\ell}(\mathbf{x}) = \mu(\mathbf{x}) + \frac{L\left[K(\cdot, \mathbf{X})\right]}{L^2[K]}(\ell - L[\mu]);$$

$$K_{f|\ell}(\mathbf{x}, \mathbf{x}') = K(\mathbf{x}, \mathbf{x}') - \frac{L\left[K(\cdot, \mathbf{X})\right] L\left[K(\mathbf{X}, \cdot)\right]}{L^2[K]}$$

- Note: This easily extends to multiple functional observations $\ell_1, \ldots, \ell_N$
- For pointwise functionals $L_x[f] = f(x)$, $\ell_i = f(\boldsymbol{x}_i)$ and it's equivalent to standard GP regression

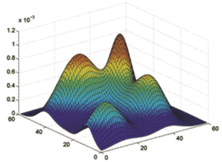
Probabilistic Numerics: Some Comments

- A new and emerging area within Probabilistic/Bayesian ML
- Applications in not just problems like computing integrals but also for
  - Numerical linear algebra problems
  - Linear/non-linear optimization (efficient solutions of linear systems of equations)
  - ODE initial value problems
  - .. and many others..
- An overview: "Probabilistic Numerics and Uncertainty in Computations" by Hennig et al (2015)
- More information and reference material: `http://probabilistic-numerics.org/`

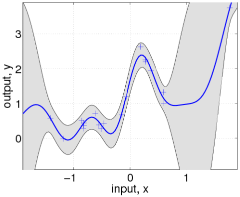
Probabilistic/Bayesian ML: The Key Takeaways

Capturing Uncertainty

- Uncertainty in the parameters learned (through the parameter's posterior distribution)

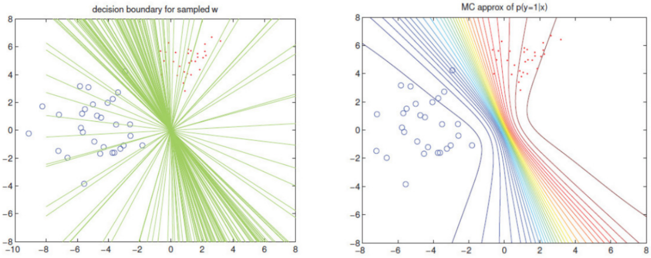


- Uncertainty in the predictions made (through the predictive posterior)



Ensemble-like Effect

- Averaging over the posterior is like using an ensemble of models

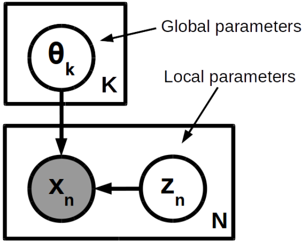


- Note: The above ensemble averages over all parameters of a single model. But, in principle, we can even make predictions by averaging over a set of different models (Bayesian model averaging)

$$p(\mathcal{D}^{(new)}|\mathcal{D}) = \sum_{m=1}^{M} p(\mathcal{D}^{(new)}|\mathcal{D}, m)p(m|\mathcal{D})$$

Generative Models

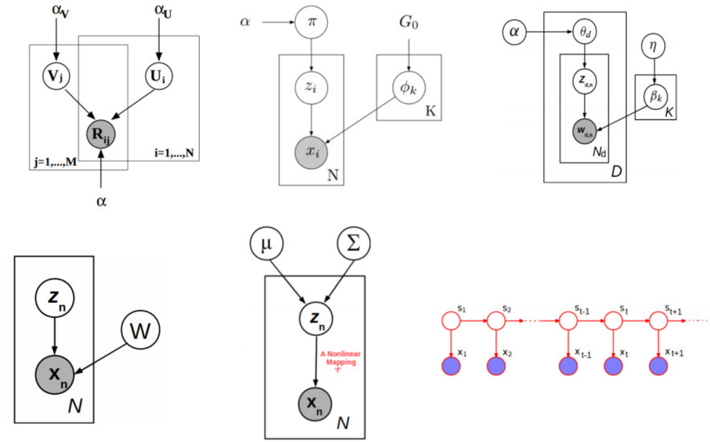
- Can model very complex data sets using generative models with latent variables
- For each observation $\boldsymbol{x}_n$, we can associate (one or more) "local" latent variables $\boldsymbol{z}_n$
- There is a set of "global" model parameters (shared by all the observations)



- The $\boldsymbol{z}$ to $\boldsymbol{x}$ mapping is problem-specific (linear/non-linear)

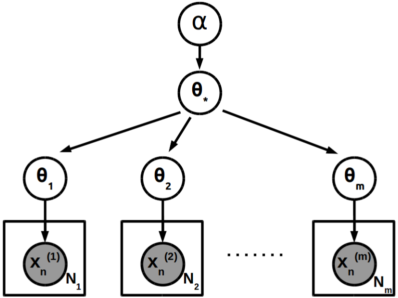
Generative Models

A variety of ML problems can be posed as generative models containing latent variables



"Modular" Construction of Complex Generative Models

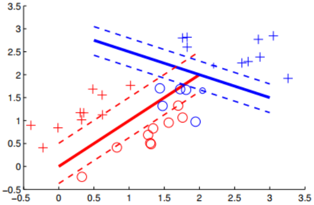
- Simple generative models can be neatly combined to design more complex models

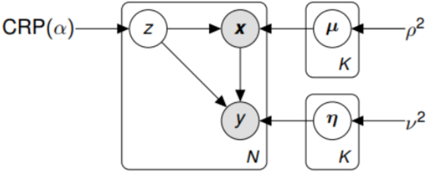


- Allows joint learning across multiple data sets (known as multitask learning or transfer learning)
- Enables different but related models to "share statistical strength"

"Modular" Construction of Complex Generative Models

- Another example: Can perform nonlinear classification using a mixture of linear classifiers
  - It is a simple yet powerful combination of two models - one that performs clustering of the data and the other that learns a linear classifier within each cluster (both learned jointly)
- Actual example shown below: DP mixture of (probabilistic) SVMs (aka "Infinite SVM", Zhu et al)





$$p_0(z_i = k|\alpha, \boldsymbol{z}_{-i}) \propto \begin{cases} n_{-i,k}, & \text{if } n_{-i,k} > 0 \\ \alpha, & \text{otherwise} \end{cases} \qquad \phi(y_i|\boldsymbol{x}_i, \boldsymbol{\eta}, z_i = k) = \exp(-2c \cdot \max(0, 1 - y_i \boldsymbol{\eta}_k^\top \boldsymbol{x}_i))$$

"Hands-Free" ML

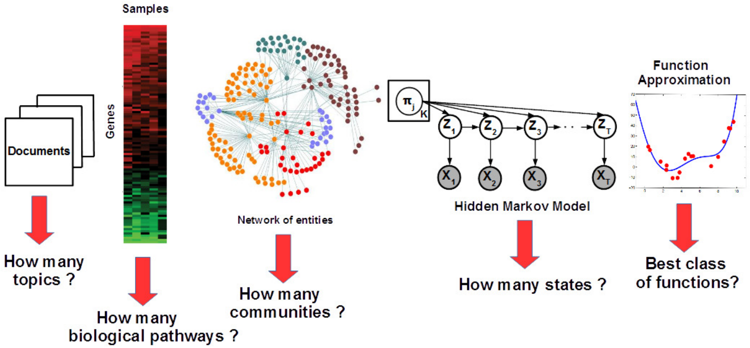
- Can learn the hyperparameters from data
  - Inferring the posterior distribution over hyperparameters
  - Inferring a point estimate of hyperparameters (Type-II MLE)
  - .. or Bayesian Optimization
- Can learn the right model complexity
  - By comparing marginal likelihood $p(\mathcal{D}|m)$ of models, e.g., by computing Bayes Factors

$$\frac{p(\mathcal{D}|m)}{p(\mathcal{D}|m')}$$

  - By constructing models having "adjustable" complexity: Nonparametric Bayesian models

Nonparametric Bayesian models

- **Nonparametric Bayesian Modeling:** A principled way to learn model size

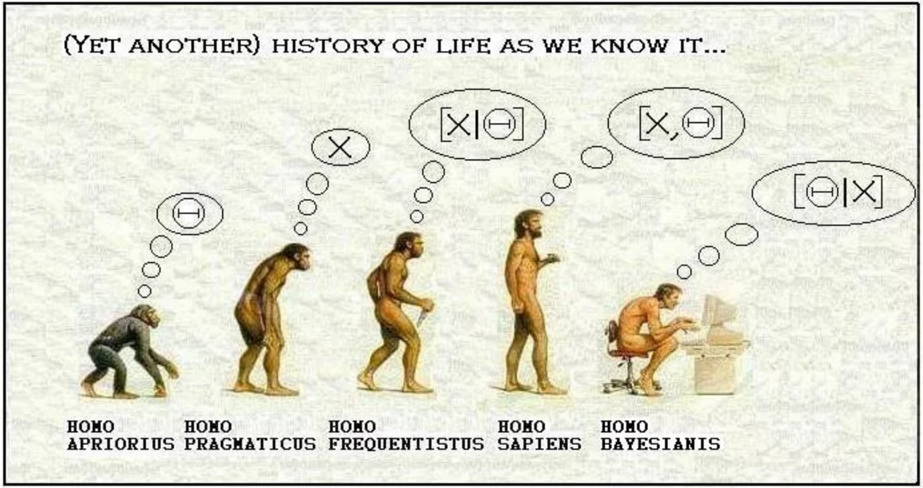


- Can be seen as an infinite limit of finite models
- The model size can grow with data (especially desirable for online settings)
- Also ideal for learning the architecture of deep learning models (# of layers and width of layer)

Some Things We Didn't Talk About

- Some other ML problems where Bayesian modeling is useful
  - (Bayesian) Reinforcement Learning
- Message-Passing algorithms for graphical models
- Theoretical results such as posterior concentration rates (roughly speaking, how quickly the posterior concentrates towards the true value of the parameter)
- Details of Probabilistic Programming (Stan, Edward, etc.)
- Philosophical debates on Bayesian vs non-Bayesian (aka Frequentist)
  - A recommended reading: "Why isn't everyone a Bayesian?" (Efron, 1986)

Thank You!


(YET ANOTHER) HISTORY OF LIFE AS WE KNOW IT...
[Θ]
[X]
[X|Θ]
[X,Θ]
[Θ|X]
HOMO APRIORIUS
HOMO PRAGMATICUS
HOMO FREQUENTISTUS
HOMO SAPIENS
HOMO BAYESIANIS