11/6/19

8805 Learning from Data: Lecture 21

Notebooks for today:

(A) gaussian-processes/Gaussian_processes_exercises.ipynb

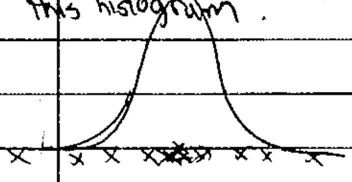
- Questions about project or mini-project IIb?

- Plan for this week and next:
  - Today:
    - Recap and complete discussion of Gaussian processes (GPs)
    - Do some exercises (not all) from (A)
    - Discuss two applications: Higdon et al. and Melendez et al.
    - Lead-in to maximum entropy
  - Friday:
    - Maximum entropy overview and class exercises
    - If time permits, function reconstruction using max ent.
  - Next week:
    - Bayesian neural networks
    - As time permits, Bayesian optimization

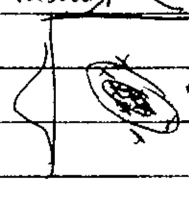
  } mini-project III

Recap of GPs

Draws from a Gaussian gives this histogram



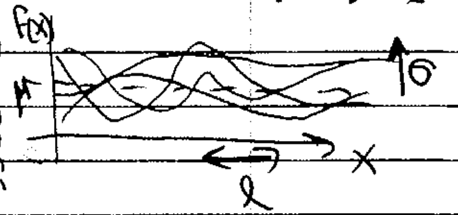
Draws from a multivariate Gaussian gives correlated posterior



$\propto e^{-(\vec{y}-\vec{\mu})^T \Sigma^{-1} (\vec{y}-\vec{\mu})}$

correlation encoded in Σ

Draws from a GP are functions $f(x) \sim GP[m(x), K(x,x')]$



- The "histogram" of GP draws will have the highest density at μ and $\approx 2/3$ within σ of μ. The GP is characterized by a kernel K (or correlation function) that gives the covariance for a multivariate Gaussian.
  - Different kernels vary in smoothness, spread, correlation length (how far apart to be uncorrelated)

e.g. $K_{RBF}(x,x') = \sigma^2 e^{-(x-x')^2/2\ell^2}$ ← very smooth

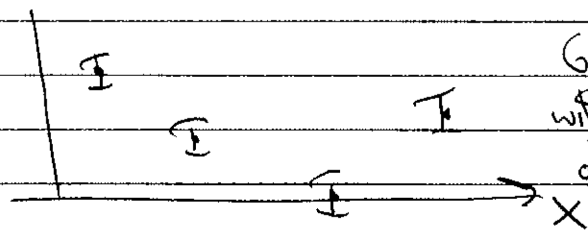
(σ^2 ← spread; ℓ: correlation)

11/6/19

training points are known precisely

uncertainties at training points

- Using GPs for interpolation or regression



Given (multidimensional) training data with errors or precise, predict at intermediate x points or extrapolate (test data)

- Impose structure through kernel.
- Here the data "speak more clearly" than for parametric regression (e.g. fitting a polynomial or a sum of gaussians ⇒ basis functions).

- Basic formulas given $\vec{\theta}$ and $\vec{x} = \begin{bmatrix} \vec{x}_1 \\ \vec{x}_2 \end{bmatrix}$ ← training N_1 pts. ← test N_2 pts

$$\begin{bmatrix} \vec{f}_1 \\ \vec{f}_2 \end{bmatrix} \Big| \vec{x}, \vec{\theta} \sim \mathcal{N}\left(\begin{bmatrix} m_1 \\ m_2 \end{bmatrix}, \begin{bmatrix} K_{11} & K_{12} \\ K_{21} & K_{22} \end{bmatrix} \right) \qquad \begin{array}{ll} K_{11} = \kappa(\vec{x}_1, \vec{x}_1) & N_1 \times N_1 \\ K_{22} = \kappa(\vec{x}_2, \vec{x}_2) & N_2 \times N_2 \\ K_{12} = \kappa(\vec{x}_1, \vec{x}_2) = K_{21}^T & N_1 \times N_2 \end{array}$$

$$\Rightarrow \vec{f}_2 | \vec{x}_2, \vec{f}_1, \vec{\theta} \sim \mathcal{N}(\tilde{m}_2, \tilde{K}_{22}) \Rightarrow \text{i.e. } p(\vec{f}_2 | \vec{x}_2, \vec{f}_1, \vec{\theta}) \text{ is a multivariate Gaussian}$$

where $\tilde{m}_2 = m_2 + K_{21} K_{11}^{-1} (\vec{f}_1 - \vec{m}_1)$ and $\tilde{K}_{22} = K_{22} - K_{21} K_{11}^{-1} K_{12}$

⇒ just plug in! Need to invert K_{11} ⇒ may be numerically unstable, so add noise $K_{11} \to K_{11} + \sigma_n^2 I_{N_1}$, even if not there previously.

- σ_n can be a parameter

<u>Calibration of a GP means determining hyperparameters $\vec{\theta} = \{\mu, \sigma, l, \sigma_n, \ldots\}$</u> given data $\vec{f} = \{f_1, f_2, \ldots, f_n\}$ at input points $\vec{x} = \{x_1, x_2, \ldots, x_n\}$ ← each is a p-dimensional vector

suppress info I here

$$\Rightarrow p(\vec{\theta} | \vec{x}, \vec{f}) \propto p(\vec{f} | \vec{x}, \vec{\theta})\, p(\vec{\theta}) \text{ by Bayes}$$

class

$\vec{m} = \mu(1, \ldots, 1)$ or more complicated

- With GP, $p(\vec{f} | \vec{x}, \vec{\theta})$ is known! $\vec{f} | \vec{x} \sim \mathcal{N}(\vec{m}, K)$ with $K = \kappa(\vec{x}, \vec{x})$
- Options: i) sample posterior by MCMC

  ii) maximize " as function of $\vec{\theta}$ by gradient descent ⇒ θ_{MAP}

  iii) for special case of conjugate priors, analytic posteriors for $\vec{\theta}$ (or part of it)

Predictions: $p(\vec{f}_2 | \vec{x}, \vec{f}_1) = \int p(\vec{f}_2 | \vec{x}, \vec{f}_1, \vec{\theta})\, p(\vec{\theta} | \vec{x}_1, \vec{f}_1)\, d\vec{\theta}$ ⇒ posterior, so distribution for $\vec{f}_2$

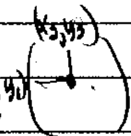
- can estimate integral with any of the options.

11/6/19

Selected exercises from (A).

1 Getting started: The Covariance Function

- start with RBF and do shift-shift-tab to see arguments
- "kern" is for kernel, another name for the covariance function
  $K(r) = \sigma^2 e^{-r^2/2l^2}$ with $r = |x_1 - x_2|$ ← stationary
- specify dimension, variance σ^2 and length scale l
- No docstring for plot ⇒ Google "Gpy plot kern"
  - x is the value to use for the 2nd argument
    ⇒ taken as 0 and then plot as function of r
- class answer Exercise 1 a)
- Do Exercise 1 b)
- Skip Covariance Functions in GPy
- Computing the Covariance Function given the Input Data, X
  - n data points in d dimensions ⇒ n×d array
  - Matern52 → recall this $K(r) = \sigma^2\left(1 + \frac{\sqrt{5}r}{l} + \frac{5r^2}{3l^2}\right) e^{-\sqrt{5}r/l}$
  - X is full of random normal ($\mu = 0$, $\sigma^2 = 1$)
  - get the covariance matrix from C = k.K(X,X)
  - $X_1 = \binom{x_1}{y_1}$, $X_2 = \binom{x_2}{y_2}$ ⇒ $r = \sqrt{(x_1-x_2)^2 + (y_1-y_2)^2}$ ↖ all combos



  - Why do we know eigenvalues are ≥ 0? (No error from log10!)
  - Notice range of eigenvalues in orders of magnitude.
  - Try Matern32 and RBF
- Given time, try combining GPs
  - Where will 2 RBF's have max? What will max be?
  - Sum is we have multiple trends in the data (e.g. a slowly changing envelope of rapidly changing behavior)

adding GPs is like an OR operation; multiplying GPs is like an AND

2 Sampling from a Gaussian Process

- Here we sample from $N(\mu, C)$ where C is Σ for X, X
- change to mu = np.linspace(-1., 1., len(X)) ⇒ underlying mean function
- Try some different covariance functions. Add

```
for i in range(nsamples):
    a.plot(X[:], Z[i,:]);
```

} What are the defaults here?

- What do you expect for nsamples = 50?

11/6/19

3 A Gaussian Process Regression Model
- Generate data + noise to fit.
- Instantiate an RBF model
- Combine with data: GPy.models.GPRegression(X, Y, k)
  - 3 parameters to optimize
  - noise is added by default. ⇒ specify noise_var in GPRegression
- Make a better fit with lengthscale = 0.1 m['rbf.lengthscale'] = 0.1
- Step Through Covariance Function Parameter Estimation

4 A Running Example
- exercise for the reader!

11/6/19

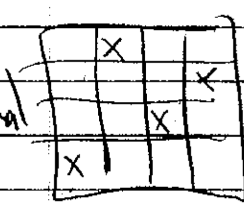
Application 1: "A Bayesian Approach for Parameter Estimation and Prediction using a Computationally Intensive Model" by Higdon et al.

⇒ Bayesian model calibration for nuclear DFT using a GP emulator
- landmark in low-energy nuclear physics but general idea of an emulator was not new.
- nuclear density functional theory (DFT): given N (neutron number) and Z (proton number), functional predicts mass of nucleus (and other properties, such as size and deformation).
  - Solve > N+Z Schrödinger equations iteratively (pairing is important)
  - For each nucleus ~5-10 minutes and want to train on about 100 nuclei ⇒ too expensive to have a model that runs the DFT for every case

⇒ - Train a GP and use this in place of the DFT model ⇒ "emulator"

- Table I shows p=12 parameters to be determined
  - PCA and SVD used to enable a reduced basis for the model
  - With 9 combinations instead of original 12, can explain 99.9% of variations. We'll come back to this at end of course.
  - Uniform priors assigned but with well informed intervals

- Need to specify initial training set ⇒ see fig. 3
  - Uses space-filling Latin hypercube ⇒ multidimensional generalization of 2D "unchallenged rook" problem



- Figure 8 shows posterior for θ ⇒ main goal

Guess:
  - What is well determined? $[E^{NM}/A]$; what returns prior? $[V_0^p]$ What pairs are highly correlated $[1/m_s^*$ and $C_0^{\rho\Delta\rho}$ or $C_0^{\rho\nabla J}$
  - An output from the posterior is a prediction for a new measurement
- Figure 10 shows how well it works. Predicted 90% intervals for $\eta(\theta)+\epsilon$ (light blue). Is it too conservative?

11/6/19

Application 2: Melendez et al. paper on EFT truncation errors.
- Rather than a particular GP being used for regression, we are most interested in learning the hyperparameters!

Go through MSU statistics_conference_2018_furnstahl_pdf.pdf
"Bayesian Statistics for Effective Field Theories"

pg. 9: analog to EFT: complete low-energy characterization; works only up to a boundary (breakdown); gets worse as boundary is approached; prior knowledge: naturalness of appropriately scaled coefficient. Why prior to take?

pg 13: what kind of statistics problem do we have?
GPs are useful for EFT truncation error.

pg. 18-29: GPs for coefficient functions (eg. in energy or scattering angle)
Plan: use low-order predictions to learn underlying GP hyperparameters; then use to predict omitted terms.
p. 30: real calculations look like this,

p. 35-39 Hierarchical statistical model
p. 40-42 GP: learn μ, σ, l. Conjugate priors mean we get results for μ, σ immediately. l stills needs to be sampled or optimized.
Note: curve-wise vs. point-wise model. Latter misses correlations.

p. 43-44 Real world error bands for nucleon-nucleon observables.

p. 45-47 Lead in to later discussion: model checking

p. 48 Physics discovery: What is the EFT breakdown scale for different observables? A new frontier!