lock into MAP w/ PyMC as easy estimate vs. full posterior

Session 7 Paper : Macke 2015   3/6/2022

2. State space models w/ linear dynamics and count-process observations

- consider models for spike data
  - record simultaneously for q neurons
  - discretized in time
  - $y_{i,t}$: spike counts for neuron $i \in \{1, \dots, q\}$ at time $t \in \{1, \dots, T\}$
    - $\mathbf{y}_t$: vector of neuron counts at time t
    - $\mathbf{y}_{1:T}$: the $q \times T$ matrix of all observations
  - $z_{i,t}$: intermediate variable for neuron i at time t
    - capture dependence of spike rate on 3 factors:
      1. param. d_i for the mean firing rate of neuron i
      2. influence of unobserved processes summarized over a p-dim. state vector $\mathbf{x}_t$
      3. any observed external covariates $\mathbf{s}_t$
    - $\mathbf{z}_t = C\mathbf{x}_t + D\mathbf{s}_t + \mathbf{d}$
      - C : $q \times p$ matrix determines how each neuron is influenced by the latent space $\mathbf{x}_t$
        - each row contains the couplings of 1 neuron to the p latent states
      - $\mathbf{s}_t$ often used to model spiking history (model refractory period) — ? D is the corresponding data
- $P(y_{i,t} \mid z_{i,t}) \sim \text{Poisson}(\eta(z_{i,t}))$ where $\eta(\cdot)$ is some non-linear fxn (eg. $\exp(\cdot)$) — standard link fxn for Pois. GLM

$$P(y_{i,t} \mid z_{i,t}) = \frac{1}{y_{i,t}!}\, \eta(z_{i,t})^{y_{i,t}}\, e^{-\eta(z_{i,t})}$$

$\mathbf{x}_1 \sim N(\mathbf{x}_0, Q_0)$ — expected initial values

$\mathbf{x}_t \mid \mathbf{x}_{t-1} \sim N(A\mathbf{x}_{t-1} + B\mathbf{u}_t, Q)$

A: "dynamics matrix"

$B\mathbf{u}_t$: "external driving input" — note difference w/ $D\mathbf{s}_t$ is their effects future instead of present

$B\mathbf{u}_t$: capture dependence of latent state on external cov.

examples:

3. Reconstructing the state from neural spike trains

- $y_{1:T}$: population data $\qquad$ $x_{1:T}$: unobserved seq. of states
- goal: use $y_{1:T}$ to reconstruct $x_{1:T}$ $\rightarrow$ $P(x_{1:T} \mid y_{1:T})$

concat. columns of $x_{1:T}$ $\rightarrow$ x as a $pT \times 1$ vector

? same for $y_{1:T}$ $\rightarrow$ y

no closed form solution for $P(x \mid y)$ so will need to approx w/ $q(x)$

- w/ $\eta(\cdot) = \exp(\cdot)$ will only have a single peak.
- ∴ will use a Gaussian approx.: $q(x) = q(x \mid \mu, \Sigma) = N(\mu, \Sigma)$

5. Results

Session 7 reading : Kernel Methods & Gaussian Processes   3/8/2022

PRML :

- 3.3.3
- 6.0 - 6.2
- 6.4 - 6.4.3

3.3.3 Equivalent kernel

prediction mean for a linear basis fxn:

$$y(\mathbf{x}, \mathbf{m}_N) = \mathbf{m}_N^T \boldsymbol{\phi}(\mathbf{x})$$

$$= \beta \boldsymbol{\phi}(\mathbf{x})^T S_N \boldsymbol{\Phi}^T \mathbf{t}$$

$$= \sum_{n=1}^{N} \beta \boldsymbol{\phi}(\mathbf{x})^T S_N \boldsymbol{\phi}(\mathbf{x}_n) t_n$$

where $$S_N^{-1} = \underbrace{S_0^{-1}}_{\text{covariance}} + \beta \boldsymbol{\Phi}^T \boldsymbol{\Phi}$$

mean of the predictive dist at a point $\mathbf{x}$ is a linear combination of the training set target variables t_n:

$$y(\mathbf{x}, \mathbf{m}_N) = \sum_{n=1}^{N} k(\mathbf{x}, \mathbf{x}_n) t_n$$

where $k(\mathbf{x}, \mathbf{x}') = \beta \boldsymbol{\phi}(\mathbf{x})^T S_N \boldsymbol{\phi}(\mathbf{x}')$ ← "smoother matrix" "equivalent kernel"

covariance b/w $y(\mathbf{x})$ & $y(\mathbf{x}')$:

$$\mathrm{cov}[y(\mathbf{x}), y(\mathbf{x}')] = \mathrm{cov}[\boldsymbol{\phi}(\mathbf{x})^T \mathbf{w}, \mathbf{w}^T \boldsymbol{\phi}(\mathbf{x}')]$$

$$= \boldsymbol{\phi}(\mathbf{x})^T S_N \boldsymbol{\phi}(\mathbf{x}')$$

$$= \beta^{-1} k(\mathbf{x}, \mathbf{x}')$$

3/8/2022

6. Kernel Methods

6.0

- kernel function : $k(x, x') = \phi(x)^T \phi(x)$
  - symmetric fxn of its arguments
  - simplest kernel fxn uses identity function : $\phi(x) = x \rightarrow k(x, x') = x^T x'$
    - "linear kernel"
- types of kernels:
  - "linear kernel" : $\phi(x) = x \rightarrow k(x,x') = x^T x'$
  - "stationary" : deal w/ difference b/w x & x' $\rightarrow k(x, x') = k(x - x')$
    ↳ b/c invariant to translations in input space
  - "homogeneous kernels" or "radial basis functions" : depend on the magnitude of the distance b/w x & x' $\rightarrow k(x, x') = k(\|x - x'\|)$

6.1 Dual Representations

- reformulate linear models in terms of "dual representation" & kernel functions arise
- consider a linear reg. model fit by minimizing a regularized sum-of-squares:

regularization constant

$$J(w) = \frac{1}{2} \sum_{n=1}^{N} \{w^T \phi(x_n) - t_n\}^2 + \frac{\lambda}{2} w^T w$$

- set gradient of $J(w)$ w.r.t w to 0, solve for w:

$$w = -\frac{1}{\lambda} \sum_{n=1}^{N} \{ w^T \phi(x_n) - t_n \} \phi(x_n)$$

$$= \sum_{n=1}^{N} a_n \phi(x_n) \quad = \underbrace{\Phi^T}_{\text{design matrix}} a \quad \text{where } a = (a_1, \ldots, a_N)^T, \; a_n = -\frac{1}{\lambda}\{w^T \phi(x_n) - t_n\}$$

- w is now a linear comb. of vectors $\phi(x_n)$
- can reformulate the least squares alg. w.r.t vector a → "dual rep."
  - substitute: $w = \Phi^T a$ into $J(w)$

$$J(a) = \frac{1}{2} a^T \Phi \Phi^T \Phi \Phi^T a - a^T \Phi \Phi^T t + \frac{1}{2} t^T t + \frac{\lambda}{2} a^T \Phi \Phi^T a$$

- def "Gram matrix" : $K = \Phi \Phi^T$
  - is a $N \times N$ symmetric matrix
  - elements of K:

$$K_{nm} = \phi(x_n)^T \phi(x_m) = k(x_n, x_m) \leftarrow \text{"kernel fxn"}$$

- w/ Gram matrix :

$$J(a) = \frac{1}{2} a^T K K a - a^T K t + \frac{1}{2} t^T t + \frac{\lambda}{2} a^T K a$$

- set gradient of $J(a)$ w.r.t a to 0, solve for a:

"target" i.e. "y"

$$a = (K + \lambda I_N)^{-1} t$$

- substitute back into linear regression model :

$$y(x) = w^T \phi(x)$$
use the other rep w/ a instead of w
$$= a^T \Phi \phi(x)$$
$$= \underbrace{k(x)^T}\underbrace{(K + \lambda I_N)^{-1}}_{\text{relation of training data}} t$$
← interpretation : pred. on new data is a weighted sum of similarity w/ training data

where vector $k(x)$ has elements : $k_n(x) = k(x_n, x)$ ← "how similar this x is to training data"

6.2 Constructing Kernels

becomes an infinitely dim. basis fxn

- a kernel fxn must correspond to a scalar product in potentially infinite dim. feature space
- necessary & sufficient conditions for fxn $k(x, x')$ to be a valid kernel :
  - the Gram matrix K must be positive semi-definite for all possible choices of x_n
- there are some properties of kernel fxns on page 296
  - can build more complex kernels from simpler ones
- Gaussian kernel:

$$k(x, x') = \exp\left(-\|x - x'\|^2 / 2\sigma^2\right)$$

6.4 Gaussian Processes (GP)

- extend the role of kernels to probabilistic discriminative models leads to the framework of GP
- instead of def. a parametric model, GP sets a prior prob. distribution over functions
  - only need to consider the values for the functions at observed values x

6.4.1 Linear regression revisited

- return to lin. reg. & re-derive the prediction dist. in terms of a dist. of functions over $y(x,w)$

our model: $y(x) = w^T \phi(x)$ where $\phi(x)$ is a lin. comb of M basis functions

M-dim weights; input x

prior over w: $p(w) = N(w \mid 0, \alpha^{-1} I)$ isotropic Gaussian

hyperparameter for precision

- prior prob. over w induces a prob. dist. over $y(x)$
- interested in joint dist. over $y(x_1), \dots, y(x_N)$ → vectorized $y = \Phi w$
  - where Φ is the design matrix w/ elements: $\Phi_{nk} = \phi_k(x_n)$
- prob. distribution for y b/c it is a lin. comb. of Gaussian variables ∴ y is Gaussian
  - ∴ just need the mean and covariance to fully define

$$E[y] = \Phi E[w] = 0 \quad \leftarrow \text{given prior for } w \sim N(0, \alpha^{-1})$$

$$\text{cov}[y] = E[yy^T] = \Phi E[ww^T] \Phi^T = \frac{1}{\alpha} \Phi \Phi^T = K \quad \leftarrow \text{Gram matrix}$$

where K is the Gram matrix: $K_{nm} = k(x_n, x_m) = \frac{1}{\alpha} \phi(x_n)^T \phi(x_m)$

- can define the kernel directly instead of through a choice of basis functions:
  - Gaussian kernel: $k(x,x') = \exp(-\|x-x'\|^2 / 2\sigma^2)$
  - exponential kernel: $k(x,x') = \exp(-\theta |x-x'|)$

6.4.2 GP for regression

- need to account for the noise in observations: $t_n = y_n + \epsilon_n$

  $y_n = y(x_n)$ — random noise

- consider noise processes w/ Gaussian dist.:

$$p(t_n | y_n) = \mathcal{N}(t_n | y_n, \beta^{-1})$$

hyperparam. for precision of noise

joint dist. for all target values $t = (t_1, \ldots, t_N)^T$ conditioned on $y = (y_1, \ldots, y_N)^T$:

$$p(t|y) = \mathcal{N}(t | y, \beta^{-1} I_N) \quad \leftarrow \text{isotropic Gaussian}$$

- marginal dist. $p(y)$ is a Gaussian w/ mean 0 and cov. K (Gram matrix)

$$p(y) = \mathcal{N}(y | 0, K)$$

- integrate out y to get marginal dist. of t:

$$p(t) = \int p(t|y) p(y) \, dy$$

$$= \mathcal{N}(t | 0, C)$$

where cov. matrix C: $C(x_n, x_m) = k(x_n, x_m) + \beta^{-1} \delta_{nm}$

randomness from $y(x)$ — randomness of ϵ

- common kernel for GP regression: "exponential of a quadratic form":

$$k(x_n, x_m) = \theta_0 \exp\left[-\frac{\theta_1}{2} \| x_n - x_m \|^2 \right] + \theta_2 + \theta_3 x_n^T x_m$$

"t" for "target" not "training"

- given training data: $\mathbf{t}_N (t_1, \dots, t_N)^T$ corresponds to $(x_1, \dots, x_N)$
- want ~~to~~ make prediction on t_{N+1} given new x_{N+1}
  - → evaluate $p(t_{N+1} \mid \mathbf{t}_N, x_1, \dots, x_N, x_{N+1}) \equiv p(t_{N+1} \mid \mathbf{t}_N)$ (for simpler notation)

- begin eval. of $p(t_{N+1} \mid \mathbf{t}_N)$ w/ joint dist. $p(\mathbf{t}_{N+1})$

$$p(\mathbf{t}_{N+1}) = \mathcal{N}(\mathbf{t}_{N+1} \mid 0, C_{N+1})$$

- partition cov. matrix C: $C_{N+1} = \begin{pmatrix} C_N & k \\ k^T & c \end{pmatrix}$

  where C_N is $N \times N$ cov. mat. for $n, m = 1, \dots, N$

  k is the vector of elements: $k(x_n, x_{N+1})$ for $n = 1, \dots, N$

  c is the scalar $k(x_{N+1}, x_{N+1}) + \beta^{-1}$

- therefore, conditional dist. $p(t_{N+1} \mid \mathbf{t}_N)$ is Gaussian w/

  mean: $m(x_{N+1}) = k^T C_N^{-1} \mathbf{t}$

  cov: $\sigma^2(x_{N+1}) = c - k^T C_N^{-1} k$

- mean of the predictive dist: $m(x_{N+1}) = \sum_{n=1}^{N} a_n k(x_n, x_{N+1})$

  where a_n is the n^{th} component of $C_N^{-1} \mathbf{t}$

Session 7 Lecture Notes 3/9/2022

1. Kernel methods

- a GP is a kernel method

2. Gaussian processes

- reason GP is possible b/c we marginalize out the unobserved values of x from the multivariate Gaussian
  - good explanation in lecture slides