Bayesian Non-Parametric Methods

Casey Gibson
UMass Amherst

August 7, 2017

WHY?

- KCDE performs very well, can we do better?
- Integration with Bayesian reporting delay framework

$$\lambda_t \sim Ga(\alpha, \beta)$$

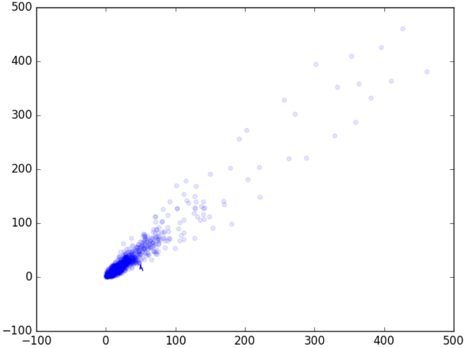
$$N(t, \infty)|\lambda_t \sim Po(\lambda_t)$$

$$N(t, T)|N(t, \infty), q_{T-t} \sim Bin(N(t, \infty), q_{T-t})$$

- Handle large number of covariates
- Estimation algorithm
  - LOO-CV
  - Backpropagation
  - HMC and ADVI

Problem Statement

- **Problem** Given a time series X_i for $i = 1..n$, can we predict X_{t+1} based on X_t?
- Transform problem into $X_t, X_t{-}1$ axis

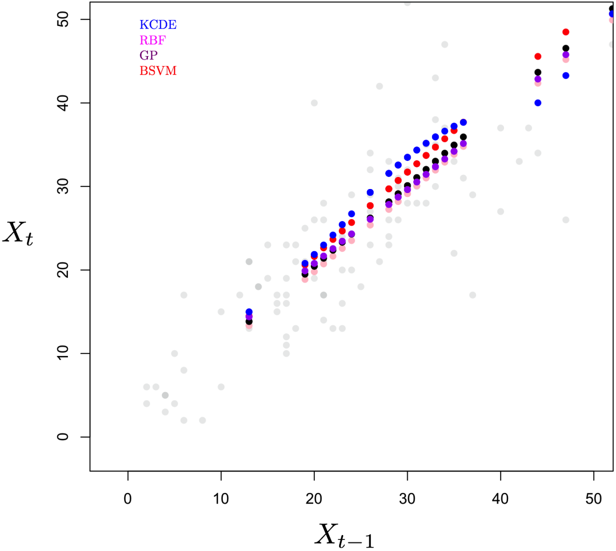


PROBLEM STATEMENT

- In this framework multiple different models were evaluated on the San Juan Dengue data with the following results

Method	50/10	100/50	MP
KCDE (NP)	29.2	64.9	2599.0
GP	34.2	62.5	2280.4
BSVM	22.9	68.0	2704.2
BNN	27.02	60.1	2350.4
RBF	17.4	61.9	2287.7

Table: Agriculture, Source: Cryer (1986), in file: data/milk, Description: Monthly milk production: pounds per cow. Jan 62 Dec 75


KCDE
RBF
GP
BSVM
X_t
X_{t-1}
0
10
20
30
40
50
0
10
20
30
40
50

OVERVIEW OF METHODS

- Gaussian Process Family
  - Pure Gaussian Process
  - Bayesian Neural Network
  - Relevance Vector Machine
- Dirichlet Process Family
  - Stick breaking density mixture model
  - Probit breaking dependent density regression

GAUSSIAN PROCESS

- **Def** Let $f : R \to R$ be a random function defined on all points $x_i \in R$. $P(f)$ is a **GP** if for any finite subset $\{x_1, \ldots X_n\}$ $P(f(x_1), f(x_2), .., f(x_n))$ has a multivariate Gaussian Distribution
$$P(f) \sim N(\mu(x), K(x_i, x_j))$$
where $\mu(x)$ is some mean function and $K(x_i, x_j)$ is some positive definite kernel function.
- **Intuition** GP defines a distribution over functions defined on R
- Why did we cover GPs first? All other methods define **distributions over functions**, just in slightly different ways
- Demo Time

GAUSSIAN PROCESS

- **Function**

$$y_i = f(x_i) + \epsilon_i$$

- **Prior**

$$\mathbf{f} \sim \mathbf{GP}(0, K(x_i, x_j))$$

- **Likelihood**

$$\begin{bmatrix} \mathbf{f} \\ \mathbf{f}_* \end{bmatrix} \sim \mathcal{N}\left(\begin{bmatrix} \boldsymbol{\mu} \\ \boldsymbol{\mu}_* \end{bmatrix}, \begin{bmatrix} \Sigma & \Sigma_* \\ \Sigma_*^\top & \Sigma_{**} \end{bmatrix} \right)$$

- **Posterior**

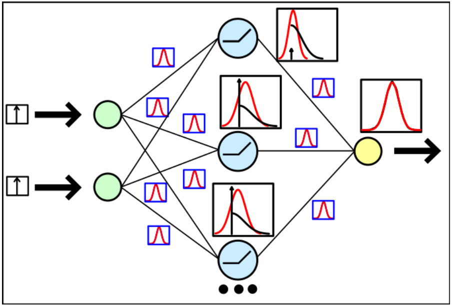
$$\mathbf{f}_*|\mathbf{f} \sim \mathcal{N}(\mu_* + \Sigma_*^\top \Sigma^{-1}(\mathbf{f} - \boldsymbol{\mu}),\ \Sigma_{**} - \Sigma_*^\top \Sigma^{-1} \Sigma_*)$$

BAYESIAN NEURAL NETWORK AS GP APPROXIMATION

- Classic NN

$$f(X) = W\sigma(b + VX)$$

- Bayeisan NN



BAYESIAN NEURAL NETWORK AS GP APPROXIMATION:MORE DETAIL

- **Claim**: An infinite Bayesian neural network converges to a GP
- **Sketch of Proof** Consider

$$f(x) = \frac{1}{K}\sum_{i=1}^{K} w_i h_i(x)$$

where w_i and $h_i(x)$ are R.V. by construction. Therefore

$$w_i h_i(x)$$

is some R.V. with a pdf with mean

$$E(w_i h_i(x)) = E(w_i)E(h_i(x)) = 0$$

. Therefore by the CLT we have that the sum converges to

$$N(0, \sigma^2 Var(h_i(x))$$

BAYESIAN NEURAL NETWORK: NEAL'S CONSTRUCTION

- **Function**

$$y_i = w_2 \cdot tanh(w_1 \cdot x_i) + \epsilon_i$$

  universal approximation theorem

- **Prior**

$$W_i \sim N(0, \sigma_i^2)$$

- **Likelihood**

$$L(w_1, w_2|D) = \prod_{i=1}^{N} N(y_i; f(x_i, w_1, w_2), \sigma_3^2)$$

[http://www.cs.toronto.edu/ radford/ftp/bbp.pdf]
[Neural Networks: a replacement for Gaussian Processes? Matthew Lilley and Marcus Frean Victoria University of Wellington]

RELEVANCE VECTOR MACHINE

- **Function**: $f(x) = \sum_{i=1}^{n} w_i \cdot \Phi(x, x_i) + w_0$
- **Prior**:

$$w_i \sim N(0, \alpha_i^{-1})$$

- **Likelihood**:

$$L(w_1, ...w_n|D) \sim \prod_{i=1}^{n} N(y_i; f(x_i), \sigma_i^2)$$

Analytic posterior available, fit hyper-parameters with EM

BAYESIAN RADIAL BASIS FUNCTION NEURAL NETWORK

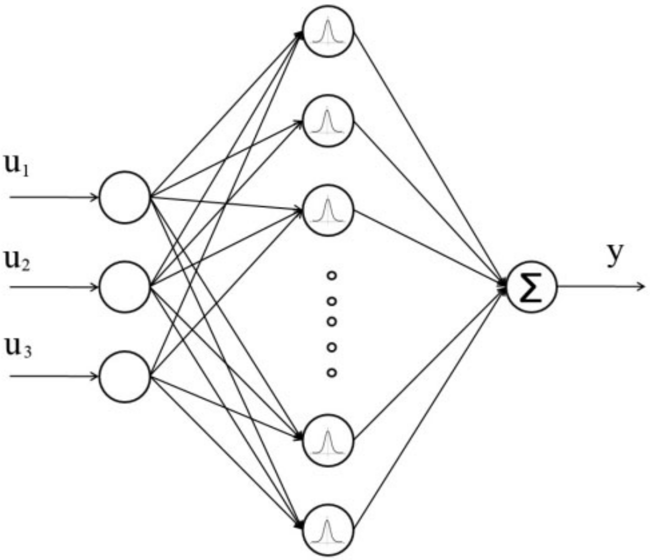
- Very similar to KCDE
- "non-parametric" Neural Network method in that the architecture size scales with the data

$$\phi(x) = \sum_{i=1}^{n} a_i \rho(||x - c_i||)$$

where

$$\rho(||x - c_i||) = exp(-\beta(x - c_i)^2)$$

Estimation is a combination of k-means and backprop


u1
u2
u3
Σ
y

BUT WAIT, WHERE IS BAYES?

- Use ADVI or HMC just as regular BNN
- We can take arbitrary NN architectures and treat them as bayesian approximations of GP using dropout
- **Intuition** Every time we pass a training example to the NN we randomly "drop" a weight (set it to zero), therefore each training example is sent through a random NN that is slightly different from the previous one. At test time we use a MC average of each NN
- We can also output Gaussian Mixture coefficients μ, σ^2 and minimize NLL

FURTHER RESEARCH

- Compare NLL scores of methods
- Which method produces the best CIs?
- Some models may be better at point prediction and some may be better at uncertainty, can we leverage both in ensemble?

REFERENCES

- A. Asvadi, M. Karami, Y. Baleghi, "Efficient Object Tracking Using Optimized K-means Segmentation and Radial Basis Function Neural Networks," International Journal of Information and Communication Technology Research (IJICT), vol. 4, no. 1, pp. 29-39, December 2011.
- Gelman, Andrew, et al. Bayesian data analysis. Vol. 2. Boca Raton, FL: CRC press, 2014.
- Herndez-Lobato J. M. and Adams R. "Probabilistic Backpropagation for Scalable Learning of Bayesian Neural Networks," In ICML, 2015.
- Murphy, Kevin P.."Machine Learning: A Probabilistic Perspective",Adaptive Computation and Machine Learning series . The MIT Press, August 2012
- Yarin Gal and Zoubin Ghahramani. "Dropout as a Bayesian approximation: Representing model uncertainty in deep learning." arXiv:1506.02142, 2015