Variational Learning for Gaussian Processes at Large Scale

Mohammad Emtiyaz Khan

EPFL, Lausanne, Switzerland

July 28, 2014

Joint GSoC work with
Heiko Strathman (UCL) and Wu Lin (Univ of Waterloo)

The Big Picture

Joint density models for data with mixed data types

Bayesian models – principled and robust approach

Algorithms that are not only accurate and scalable, but are also easy to tune and implement.

Today's talk

Gaussian processes (GPs)

Variational algorithm

Gaussian Processes at Large Scale

- What are GPs? See Shogun Notebook on GPs!

  http://www.shogun-toolbox.org/static/notebook/current/gaussian_processes.html

- Why GPs? For me, for uncertainty quantification.

  Applications: Recommendation systems, black-box optimization.

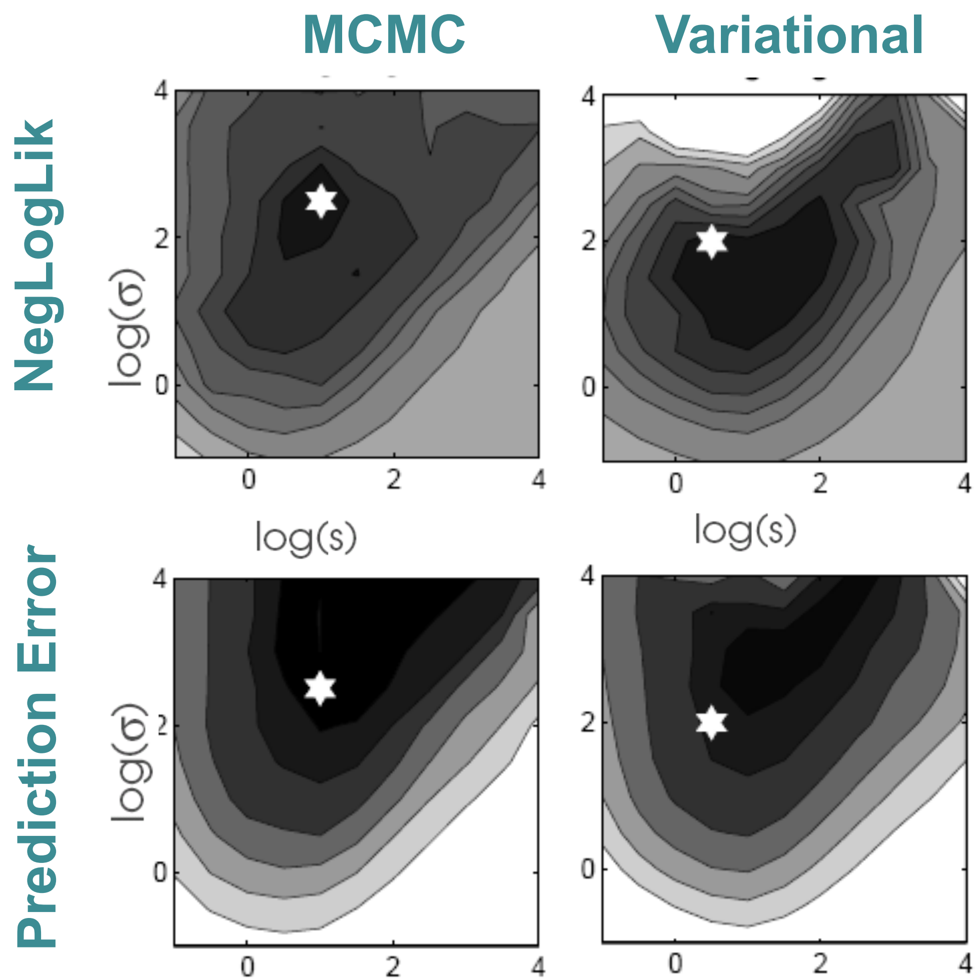
- Why large scale?
- Problems: Scalability problems due to big covariances
  - Inversion, storage, and optimization.
- Variational method is a promising solution.

Mohammad Emtiyaz Khan

MCMC vs Variational

Multiclass Gaussian Process: Glass dataset (N=143, K=6) (Khan et. al. Aistats 2012)

2 months



3 days

Mohammad Emtiyaz Khan

MCMC and its Scalability

On Netflix (for latent factor model with Gaussian likelihood)

	Time/iter (parallel)	Total Iters	Total Time	RMSE
MAP (D=100)	5 Mins	30	2.5 Hrs	0.93-0.90
Gibbs (D=30)	11 Mins	800	≈ 6 Days	0.89
Variational (D=30)	10 Mins	30	≈ 5 Hrs	0.89

Taken from Zhou et.al. 2010, Salakhutdinov & Mnih 2008

Outline

- Variational Gaussian approximation.
- A rough overview of algorithms.
- Future plans for large-scale implementation.
- Ipython notebook (GsoC 2014 code done by Wu Lin).

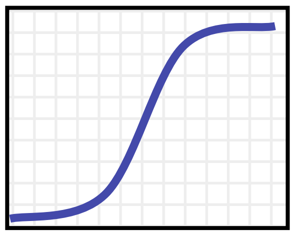
Variational Learning for GPs

$$\log \int \left[\prod_{n=1}^{N} p(y_n|z_n,\theta_y)\right] \mathcal{N}(\mathbf{z}|\mu_X, \mathbf{\Sigma}_X)d\mathbf{z}$$

Variational Gaussian approximation

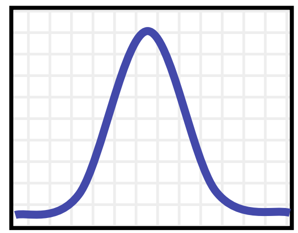
$\log \int \prod_{n=1}^{N}$

Bernoulli-logit



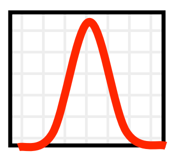
x

Prior



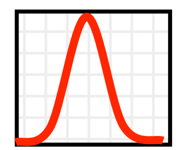
$d\mathbf{z}$

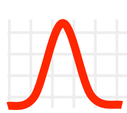
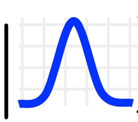
Posterior



x

Posterior



$\geq \max_{\mathbf{m},\mathbf{V}} \; -D_{KL}\Big($  $\|$  $\Big) + \sum_{n=1}^{N} \int \log p(y_n|z_n,\theta_y)\mathcal{N}(z_n|m_n,V_{nn})dz_n$

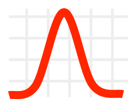
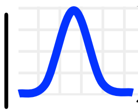
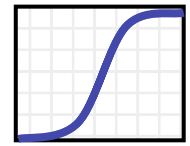
Optimization of the lower bound

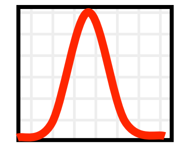
NIPS 2012, ICML 2013, AISTATS 2014, NIPS 2014

Local bounds

NIPS 2010, ICML 2011, AISTATS 2012

Variational Gaussian Approximation

$\geq \max_{\mathbf{m},\mathbf{V}} \; -D_{KL}\Big($  $\Big|\Big|$  $\Big) + \sum_{n=1}^{N} \int \log$ 



- MAP assumes ***V*** = 0 (convex).
- Mean-Field (MF) assumes ***V*** is diagonal (convex).
- Cholesky method optimizes wrt Cholesky of ***V*** (convex).
- Kullback-Leibler (KL) method reparameterizes ***V*** (but not convex)

$$\mathbf{V} = (\boldsymbol{\Sigma}^{-1} + \mathrm{diag}(\boldsymbol{\lambda}))^{-1}$$

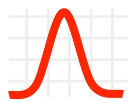
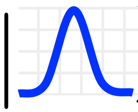
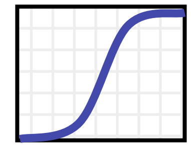
- Dual method, similar as KL, but convex for some classes.

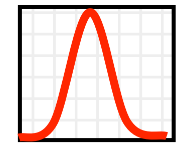
Less accurate — More accurate

Accuracy	MAP < MF < Dual < Decoupled = Cholesky = KL
Computation	MAP < MF < Dual = Decoupled < Cholesky ? KL

Less computation — More computation

Decoupled Variational Inference

$\geq \max_{\mathbf{m},\mathbf{V}} -D_{KL}$ ( || ) $+ \sum_{n=1}^{N} \int \log$ 



$$\text{s.t. } \underbrace{h_n = m_n}_{\alpha_d}, \underbrace{\rho_n^2 = V_{nn}}_{\lambda_d}$$

GP Classification using GP Regression

Repeat

For all *n*, Given ***α*** and ***λ*,** find yPseudo (O(N))

predict yPseudo using GP regression (1 Inversion)

For all *n*, compute (stochastic) gradient in parallel (O(N)).

Until convergence

A Rough Overview of Existing Methods

Less accurate | More accurate

Less computation | More computation

	MAP	MF	VG	EP	INLA	MCMC
Generality	Good	Good	Ok	Ok	Ok	Good
Accuracy	Bad	Ok	Good	Good	Good	Good
Speed	Good	Good	Ok	Ok	Ok	Bad
Scalability	Good	Ok	Ok	Ok	Bad	Bad
Implement	Good	Ok	Ok	Ok	Ok	Ok

Green	Good
Yellow	Ok
Red	Bad

MAP is maximum a-posteriori

MF is Mean field

VG is Variational Gaussian

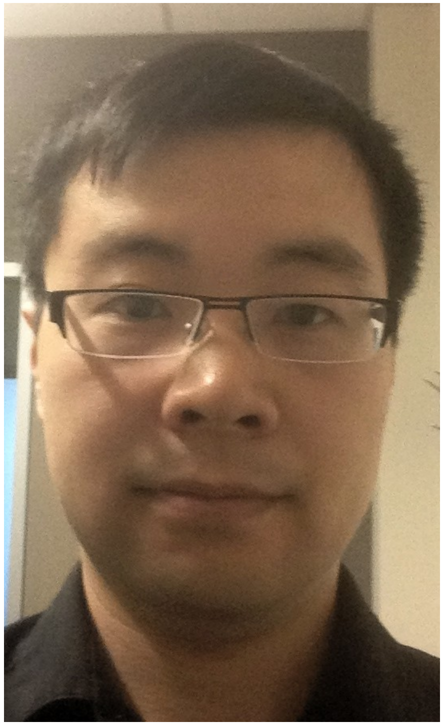
EP is Expectation Propagation

INLA is Integrated Nested Laplace

MCMC is Monte Carlo Markov Chain

[Ipython Notebook]

Wu Lin
(University of Waterloo)



Thanks
Questions?