Variational Inference in Gaussian Processes for non-linear time series

Carl Edward Rasmussen

Workshop on Nonlinear System Identification Benchmarks

Brussels, Belgium, April 25-27, 2016

with many people from the machine learning group including:

Thang Bui, Yutian Chen, Roger Frigola, Rowan McAllister, Andrew McHutchon, Rich Turner and Mark van der Wilk

Motivation

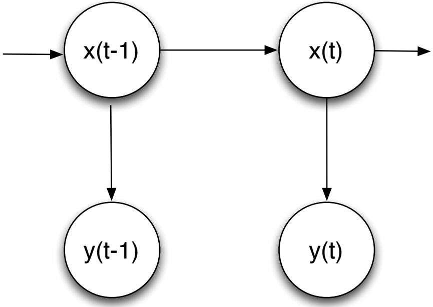
Many control problems require that the model of the dynamics be partially or entirely derived from measurements.

Therefore, the dynamics model must be stochastic, to reflect the inevitable lack of certainty in model predictions.

Flexible models are required to model a wide variety of dynamics.

We seek *automatic* model inference and training (fitting) algorithms.

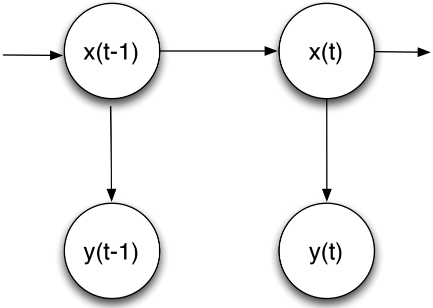
An inconvenient truth



Gaussian processes (GP) are an *extremely powerful framework* for learning and inference about non-linear functions.

Irritating fact: Although desirable, it is not easy to apply GPs to latent variable time series.

Non-linear transition model is enough

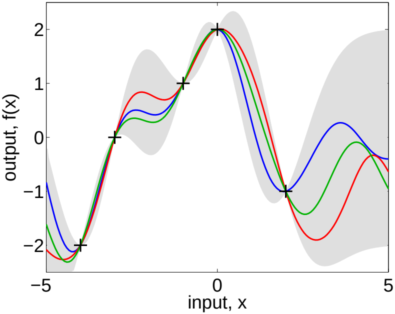


It is the non-linearity in the *transition* model which is essential.

A non-linear *observation model* can be moved to the transition model without loss of generality (but possibly at the cost of needing additional state coordinates).

Thus, in the remainder of the talk, we focus on non-linear transition models.

Gaussian Process



Gaussian processes (GP) are flexible stochastic models.

A GP specifies a conditional joint over (any number of) function values, given the inputs (index set) $p(f(x_1), \ldots, f(x_n)|x_1, \ldots, x_n)$.

A GP is a non-parametric model; the 'parameters' of the model is the function itself. This creates some interesting opportunities and some extra challenges.

Generative Model

Gaussian Process State Space Model

$$\begin{aligned} f(\mathbf{x}|\theta_x) &\sim \mathcal{GP}\big(m(\mathbf{x}), k(\mathbf{x}, \mathbf{x}')\big), && (1)\\ \mathbf{x}_t|\mathbf{f}_t &\sim \mathcal{N}(\mathbf{x}_t|\mathbf{f}_t, Q), \qquad Q \text{ diagonal}, && (2)\\ \mathbf{y}_t|\mathbf{x}_t &\sim \mathcal{N}(\mathbf{y}_t|C\mathbf{x}_t, R), && (3) \end{aligned}$$

with hyperparameters $\theta = (\theta_x, Q, C, R)$.

The joint probability is the prior times the likelihood

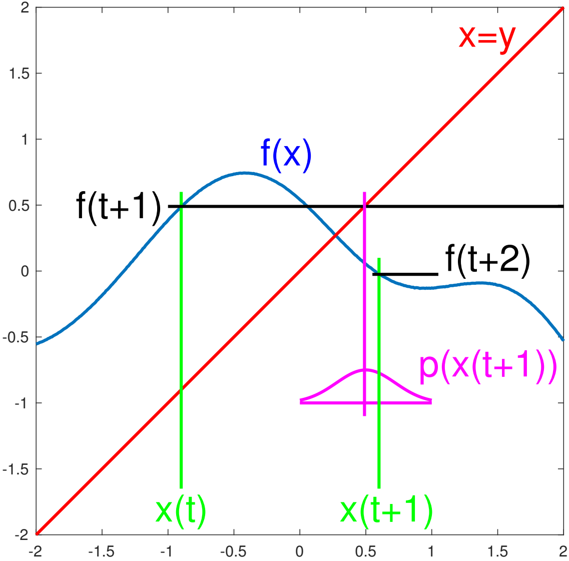
$$p(\mathbf{y}, \mathbf{x}, \mathbf{f}|\theta) = p(\mathbf{x}_0) \prod_{t=1}^{T} p(f_t|\mathbf{f}_{1:t-1}, \mathbf{x}_{0:t-1}, \theta_x) p(x_t|f_t, Q) p(y_t|x_t, C, R). \qquad (4)$$

Note that the joint distribution of the $\mathbf{f}$ values isn't even Gaussian! The marginal likelihood is

$$p(\mathbf{y}|\theta) = \int p(\mathbf{y}, \mathbf{x}, \mathbf{f}|\theta) d\mathbf{f} d\mathbf{x}. \qquad (5)$$

That's *really* awful! But we need it to train the model.

Picture of part of generative process



Let's lower bound the marginal likelihood

$$\begin{aligned}\log p(\mathbf{y}|\theta) &\geqslant \int q(\mathbf{x},\mathbf{f}) \log \frac{p(\mathbf{x}_0)\prod_{t=1}^{T} p(f_t|\mathbf{f}_{1:t-1},\mathbf{x}_{0:t-1})p(x_t|f_t)p(y_t|x_t)}{q(\mathbf{x},\mathbf{f})} d\mathbf{x}d\mathbf{f} \\ &= \int q(\mathbf{x},\mathbf{f}) \log \frac{p(\mathbf{y}|\theta)p(\mathbf{x},\mathbf{f}|\mathbf{y},\theta)}{q(\mathbf{x},\mathbf{f})} d\mathbf{x}d\mathbf{f} \\ &= \log p(\mathbf{y}|\theta) - \mathcal{KL}(q(\mathbf{x},\mathbf{f})||p(\mathbf{x},\mathbf{f}|\mathbf{y},\theta)).\end{aligned} \qquad (6)$$

for any distribution $q(\mathbf{x},\mathbf{f})$, by Jensen's inequality.

Let's chose the $q(\mathbf{x},\mathbf{f})$ distribution within some restricted family to maximize the lower bound or equivalently minimize the $\mathcal{KL}$ divergence.

This is still nasty because of the annoying prior, unless we chose:

$$q(\mathbf{x},\mathbf{f}) = q(\mathbf{x})\prod_{t=1}^{T} p(f_t|\mathbf{f}_{1:t-1},\mathbf{x}_{0:t-1}).$$

This choice makes the lower bound simple but terribly loose! Why? Because the approximating distribution over the $\mathbf{f}$'s doesn't depend on the observations $\mathbf{y}$.

Augment GP with inducing variables

Augment the model with an additional set of input output pairs $\{\mathbf{z}_i, u_i | i = 1, \dots M\}$

Joint distribution:

$$p(\mathbf{y}, \mathbf{x}, \mathbf{f}, \mathbf{u} | \mathbf{z}) \ = \ p(\mathbf{x}, \mathbf{f} | \mathbf{u}) p(\mathbf{u} | \mathbf{z}) \prod_{t=1}^{T} p(y_t | x_t). \tag{7}$$

Consistency (or the marginalization property) of GPs ensures that it is straight forward to augment with extra variables.

This step seemingly makes our problem *worse*, because we have more latent variables.

Lower bound revisited

Lower bound on marginal likelihood

$$\log p(\mathbf{y}|\theta) \;\geqslant \int q(\mathbf{x},\mathbf{f},\mathbf{u}) \log \frac{p(\mathbf{u})p(\mathbf{x}_0)\prod_{t=1}^{T} p(f_t|\mathbf{f}_{1:t-1},\mathbf{x}_{0:t-1},\mathbf{u})p(x_t|f_t)p(y_t|x_t)}{q(\mathbf{x},\mathbf{f},\mathbf{u})} d\mathbf{x}d\mathbf{f}d\mathbf{u}, \tag{8}$$

for any distribution $q(\mathbf{x},\mathbf{f},\mathbf{u})$, by Jensen's inequality.

Now chose

$$q(\mathbf{x},\mathbf{f},\mathbf{u}) \;=\; q(\mathbf{u})q(\mathbf{x})\prod_{t=1}^{T} p(f_t|\mathbf{f}_{1:t-1},\mathbf{x}_{0:t-1},\mathbf{u}). \tag{9}$$

This choice conveniently makes the troublesome $p(f_t|\mathbf{f}_{1:t-1},\mathbf{x}_{0:t-1},\mathbf{u})$ term cancel. But the $\mathbf{u}$ values can be chosen (via $q(\mathbf{u})$) to reflect the observations, so the bound may be tight(er).

The lower bound is $\mathcal{L}(\mathbf{y}|q(\mathbf{x}),q(\mathbf{u}),q(\mathbf{f}|\mathbf{x},\mathbf{u}),\theta)$.

What do we get for all our troubles?

- For certain choices of covariance function, **f** can be integrated out

$$\mathcal{L}(\mathbf{y}|q(\mathbf{x}), q(\mathbf{u}), \theta) = \int \mathcal{L}(\mathbf{y}|q(\mathbf{x}), q(\mathbf{u}), q(\mathbf{f}|\mathbf{x}, \mathbf{u}), \theta) d\mathbf{f} \tag{10}$$

- The optimal $q(\mathbf{u})$ is found by calculus is variations and turns out to be Gaussian, and can be maxed out

$$\mathcal{L}(\mathbf{y}|q(\mathbf{x}), \theta) = \max_{q(\mathbf{u})} \mathcal{L}(\mathbf{y}|q(\mathbf{x}), q(\mathbf{u}), \theta) \tag{11}$$

- The optimal $q(\mathbf{x})$ has Markovian structure; making a further Gaussian assumption, the lower bound can be evaluated analytically, as a function of its parameters μ_t, $\Sigma_{t,t}$ and $\Sigma_{t-1,t}$.

Algorithm: optimize the lower bound $\mathcal{L}(\mathbf{y}|q(\mathbf{x}), \theta)$ wrt the parameters of the Gaussian $q(\mathbf{x})$ and the remaining parameters θ (we need derivatives for the optimisation).

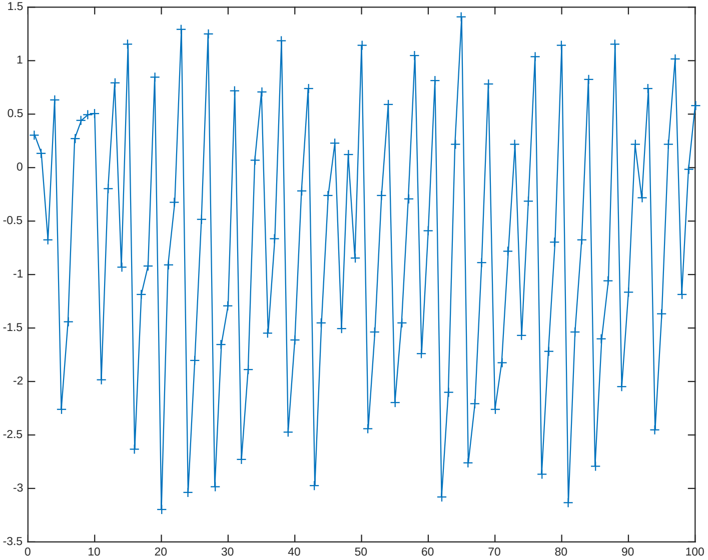
Sparse approximations are builtin

The computational cost is dominated by the GP. But, the effective ’training set size’ for the GP is given by M the number of inducing variables.

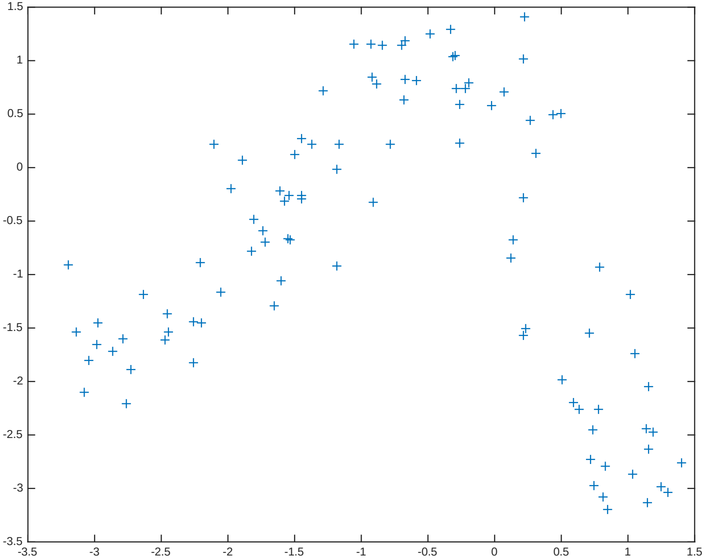
Therefore, we can chose M to trade off accuracy and computational demand.

The computational cost is linear in the length of the time-series.

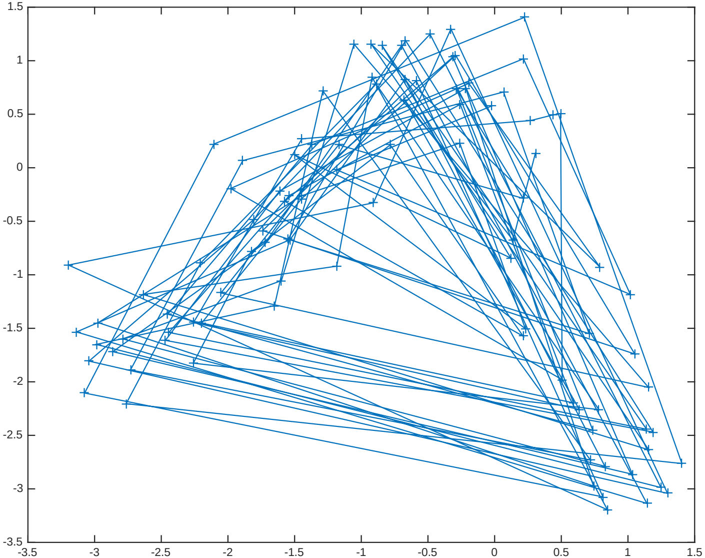
Data from non-linear Dynamical System



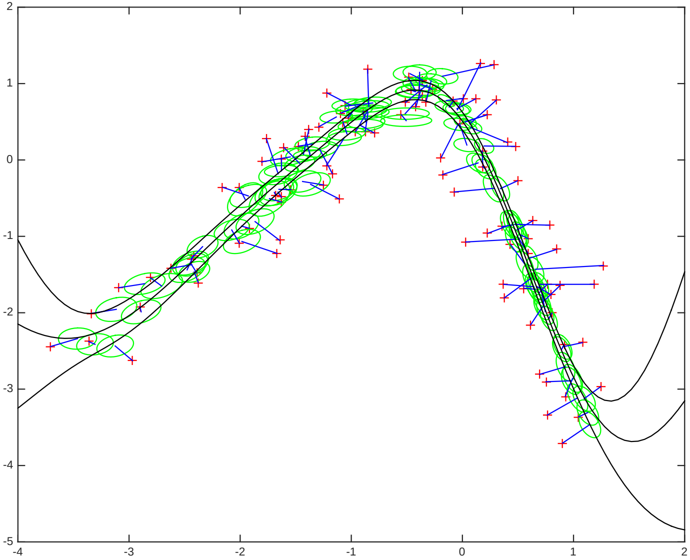
State space representation: f(t) as a function of f(t-1)



With explicit transitions



Data from non-linear Dynamical System



Parameterisation

Let's use the method on a sequence of length T, with D dimensional observations and a D dimensional latent state, ie we have TD observations.

The lower bound is to be optimized wrt all the parameters

- the pairwise marginal Gaussian distribution $q(\mathbf{x})$. This contains TD parameters for the mean and $2TD^2$ for the covariances.
- the inducing inputs z. For each of the D GPs there are MD inducing inputs, ie a total of MD^2.
- parameters of the observation model C, there are D^2 of these
- noise parameters, $2D$.
- GP hyperparameters, $\sim D^2$.

for a grand total of roughly $(2T + M)D^2$.

Example: cart and inverted pendulum, $D = 4$ and 10 timeseries each of length 100, so $T = 1000$ and $M = 50$. So we have 4000 observations and 36832 free parameters. This large number of parameters may be inconvenient but it doesn't lead to overfitting!

Implementation

Careful implementation is necessary

- $q(\mathbf{x})$ is assumed Markovian. Thus the *precision* matrix is block tridiagonal. The covariance is full, don't write it out, it's huge!
- $q(\mathbf{x})$ has to be parameterised carefully to allow all pos def matrices, but without writing out the covariances.
- initialization of parameters is important
    - most of the free parameters are in the covariance of $q(\mathbf{x})$. Initially train with *shared* covariances across time.
    - then continue training with free covariances.

Conclusions

- Principled, approximate inference in flexible non-linear non-parametric models for time series are possible and practical
- Allows to integrate over dynamics
- Automatically discover dimensionality of the hidden space – it is not the case that more latent dimensions lead to higher marginal likelihood

Some other interesting properties include

- Handles missing variables
- Multivariate latent variables and observations straight forward
- flexible non-parametric dynamics model
- Occam's Razor automatically at work, no overfitting
- Framework includes computational control by limiting M

Marginalize, don't optimize

Model with parameters θ

$$y = f_\theta(x) + \varepsilon$$

Training data set

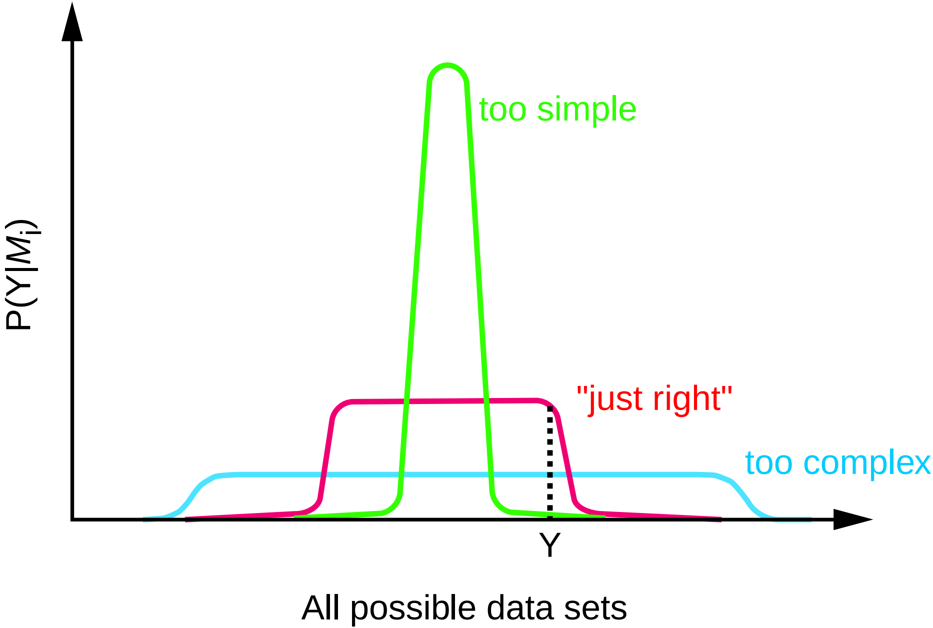
$$\mathcal{D} = \{x_n, y_n\},\ n = 1, \ldots, N$$

Make a prediction about the output y^* given a new (test) input x^*:

$$\begin{aligned} p(y^*|x^*, \mathcal{D}) &= \int p(y^*, \theta|x^*, \mathcal{D}) d\theta \\ &= \int p(y^*|x^*, \theta) p(\theta|\mathcal{D}) d\theta, \end{aligned}$$

where $p(\theta|\mathcal{D})$ is the posterior (proportional to the prior times the likelihood).

Notice, no optimization, 'estimation' or 'fitting' or 'tuning' or 'adapting', etc.


P(Y|M_i)
too simple
"just right"
too complex
Y
All possible data sets

- Bayesian inference is not generally equivalent to regularization.
- Maximising likelihood times prior is Penalized Maximum Likelihood
- problem: integrals are difficult.