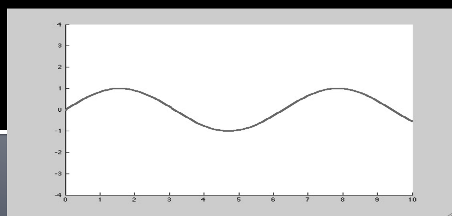
CSE-571: Gaussian Processes for Bayesian Filtering



High-level Idea of GPs

- Non-parametric regression model
- Distribution over functions
- Fully specified by training data and kernel function
- Output variables are jointly Gaussian
- Covariance given by distance of inputs in kernel space

Picture from [Bishop: Pattern Recognition and Machine Learning, 2006]

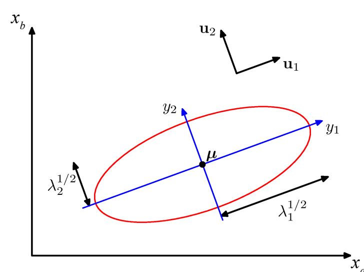
Gaussians

$$p(\mathbf{x}) = N(\mu, \Sigma)$$

$$\mathbf{x} = \begin{pmatrix} x_a \\ x_b \end{pmatrix}, \quad \mu = \begin{pmatrix} \mu_a \\ \mu_b \end{pmatrix}$$

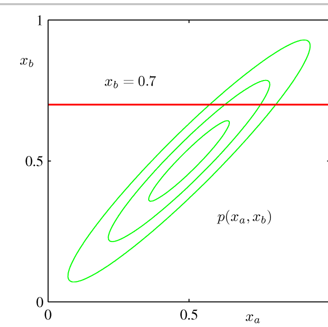
$$\Sigma = \begin{pmatrix} \Sigma_{aa} & \Sigma_{ab} \\ \Sigma_{ba} & \Sigma_{bb} \end{pmatrix}$$

$$p(\mathbf{x}) = \frac{1}{(2\pi)^{d/2} |\Sigma|^{1/2}} e^{-\frac{1}{2}(\mathbf{x}-\mu)^T \Sigma^{-1} (\mathbf{x}-\mu)}$$

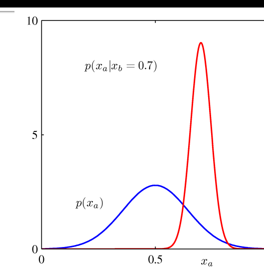


Pictures from [Bishop: PRML, 2006]

Marginalization / Conditioning



$$p(x_a) = N\left(\mu_a, \Sigma_{aa}\right)$$



$$p(x_a \mid x_b) = N\left(\mu_{a|b}, \Sigma_{a|b}\right)$$

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

GP Setting

- Outputs are noisy function of inputs:
  $y_i = f(\mathbf{x}_i) + \varepsilon$
- GP prior: Outputs jointly zero-mean Gaussian:

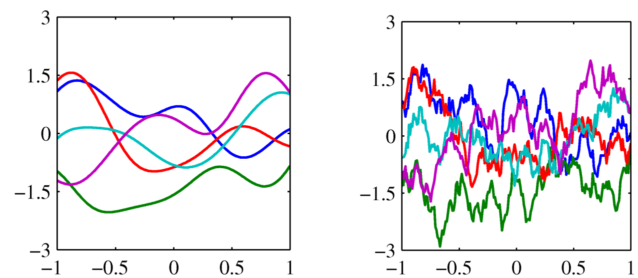
$$p(\mathbf{y} \mid \mathbf{X}) = \mathrm{N}(\mathbf{0}, \mathbf{K} + \sigma_n^2 \mathbf{I})$$

- Covariance given by kernel matrix over inputs:

$$\mathbf{K} = \begin{pmatrix} k(x_1,x_1) & \dots & k(x_1,x_n) \\ k(x_2,x_1) & & \\ \vdots & k(x_i,x_j) & \vdots \\ k(x_n,x_1) & \dots & k(x_n,x_n) \end{pmatrix} \qquad k(x,x') = \sigma_f^2 \, e^{-\frac{1}{2}(x-x')W(x-x')^T}$$

Pictures from [Bishop: PRML, 2006]

Functions Sampled from Prior



GP Prediction

- Training data:

$$D = \{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \dots, (\mathbf{x}_n, y_n)\} = (\mathbf{X}, \mathbf{y})$$

- Prediction given training samples:

$$p(y^* \mid \mathbf{x}^*, \mathbf{y}, \mathbf{X}) = N\left(\mu, \sigma^2\right)$$

$$\mu = \mathbf{k}^{*T} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{y}$$

$$\mathbf{k}^*[i] = k(\mathbf{x}^*, \mathbf{x}_i)$$

$$\sigma^2 = k(\mathbf{x}^*, \mathbf{x}^*) - \mathbf{k}^{*T} \left(\mathbf{K} + \sigma_n^2 \mathbf{I}\right)^{-1} \mathbf{k}^*$$

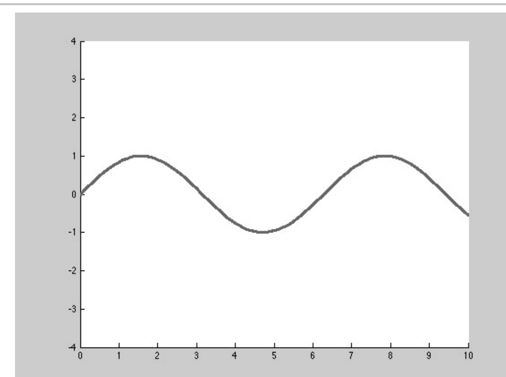
$$p(x_a \mid x_b) = \mathrm{N}\left(\mu_{a|b}, \Sigma_{a|b}\right)$$

$$\mu_{a|b} = \mu_a + \Sigma_{ab}\Sigma_{bb}^{-1}(x_b - \mu_b)$$

$$\Sigma_{a|b} = \Sigma_{aa} - \Sigma_{ab}\Sigma_{bb}^{-1}\Sigma_{ba}$$

Recall conditional

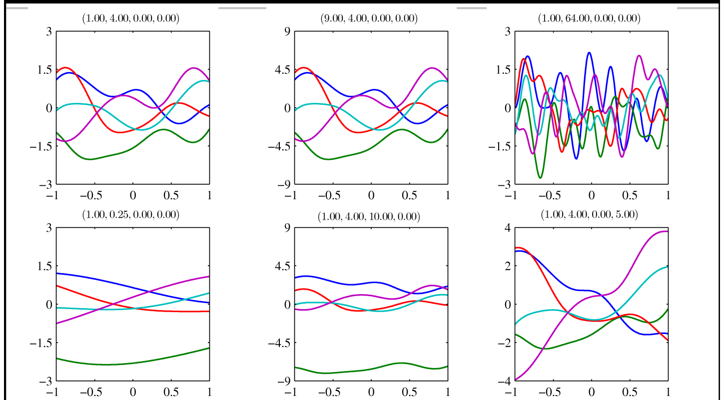
GP Prediction



Hyperparameters

Pictures from [Bishop: PRML, 2006]

$$k(x,x') = \theta_0 \exp\left(-\frac{\theta_1}{2}\|x-x'\|^2\right) + \theta_2 + \theta_3 x^T x'$$



Hyperparameter Estimation

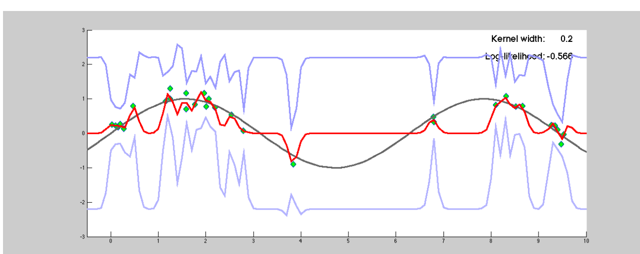
- Maximize data log likelihood:

$$\theta_* = \arg\max_{\theta} p(\mathbf{y} \mid \mathbf{X}, \theta)$$

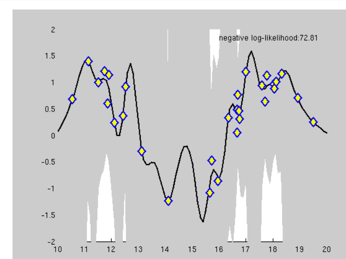
$$\log p(\mathbf{y} \mid \mathbf{X}, \theta) = -\frac{1}{2}\mathbf{y}^T\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right)^{-1}\mathbf{y} - \frac{1}{2}\log\left(\mathbf{K}+\sigma_n^2\mathbf{I}\right) - \frac{n}{2}\log 2\pi$$

- Compute derivatives wrt. params $\theta = \langle \sigma_n^2, l, \sigma_f^2 \rangle$
- Optimize using conjugate gradient descent

Kernel Width

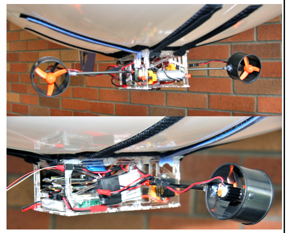


GP Optimization



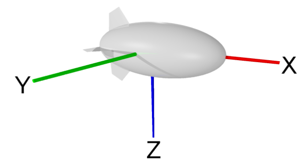
- Learn hyperparameters via numerical methods
- Learn noise model at the same time

Blimp Platform



- System:
  - Commercial blimp envelope with custom gondola
  - XScale based computer with Bluetooth connectivity
  - Two main motors with tail motor (3D control)
- Ground truth obtained via VICON motion capture system

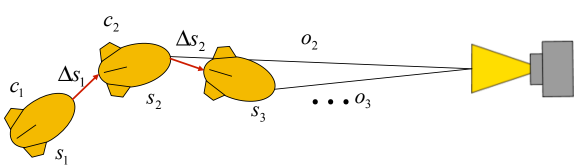
Non-linear Parametric Model



$$\dot{s} = \frac{d}{dt}\begin{bmatrix}p\\ \xi\\ v\\ \omega\end{bmatrix} = \begin{bmatrix}R_b^e v\\ H(\xi)\\ M^{-1}(\sum Forces - \omega * Mv)\\ J^{-1}(\sum Torques - \omega * J\omega)\end{bmatrix}$$

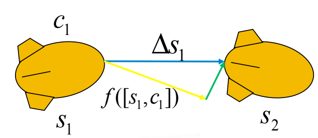
- 12-D state=[pos,rot,transvel,rotvel]
- Describes evolution of state as ODE
- Forces / torques considered: buoyancy, gravity, drag, thrust
- 16 parameters are learned by optimization on ground truth motion capture data

Learning GP Dynamics Model



- Use ground truth state to extract:
  - Dynamics data
    $D_S = \langle [s_1, c_1], \Delta s_1 \rangle, \langle [s_2, c_2], \Delta s_2 \rangle \ \ldots$
- Learn model using Gaussian process regression
  - Learn process noise inherent in system

Learning Enhanced-GP Models



- Combine GP model with parametric model
  $D_X = \langle [s_1, c_1], \Delta s_1 - f([s_1, c_1]) \rangle$
- Advantages
  - Captures aspects of system not considered by parametric model
  - Learns noise model in same way as GP-only models
  - Higher accuracy for same amount of training data

GP Modeling Accuracy

Dynamics model error

Propagation method	pos(mm)	rot(deg)	vel(mm/s)	rotvel(deg/s)
Param	3.3	0.5	14.6	1.5
GPonly	1.8	**0.2**	9.8	**1.1**
EGP	**1.6**	**0.2**	**9.6**	1.3

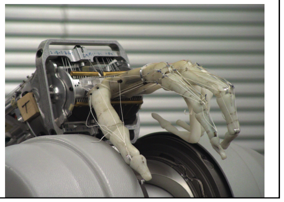
- 1800 training points, mean error over 900 test points
- For dynamics model, 0.25 sec predictions

Related Issues

- Heteroscedastic (state dependent) noise
- Non-stationary GPs
- Coupled outputs
- Sparse GPs
  - Online: Decide whether or not to accept new point
  - Remove points
  - Optimize small set of points
- Classification
  - Laplace approximation
  - No closed-form solution, sampling

Summary

- GPs provide flexible modeling framework
- Take data noise and uncertainty due to data sparsity into account
- Combination with parametric models increases accuracy and reduces need for training data
- Computational complexity is a key problem



Some References

- Website: http://www.gaussianprocess.org/
- GP book: http://www.gaussianprocess.org/gpml/
- GPLVM: http://www.cs.man.ac.uk/~neill/gplvm/
- GPDM: http://www.dgp.toronto.edu/~jmwang/gpdm/
- Bishop book: http://research.microsoft.com/en-us/um/people/cmbishop/prml/