Transfer Learning and Multiple Output Kernel Functions

Neil D. Lawrence
includes work with Mauricio Alvarez, David Luengo and Michalis Titsias

Transfer Learning for Structured Data Workshop, Whistler

12th December 2009

Outline

Outline

Styles of Machine Learning

Background: interpolation is easy, extrapolation is hard

- Urs Hölzle keynote talk at NIPS 2005.
  - Emphasis on massive data sets.
  - Let the data do the work—more data, less extrapolation.
- Alternative paradigm:
  - Very scarce data: computational biology, human motion.
  - How to generalize from scarce data?
  - Need to include more assumptions about the data (e.g. invariances).

General Approach

Broadly Speaking: Two approaches to modeling

data modeling | *mechanistic modeling*

General Approach

Broadly Speaking: Two approaches to modeling

data modeling

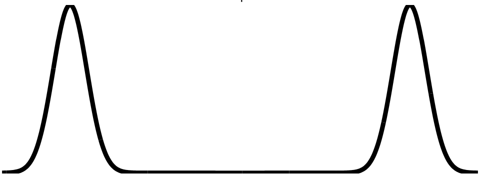
let the data "speak"

mechanistic modeling

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data “speak”	impose physical laws



General Approach

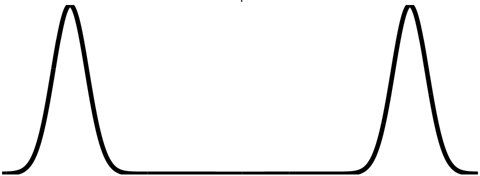
Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data “speak”	impose physical laws
data driven	

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven



General Approach

Broadly Speaking: Two approaches to modeling

data modeling

let the data “speak”

data driven

adaptive models

mechanistic modeling

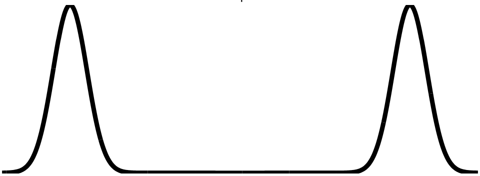
impose physical laws

knowledge driven

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations



General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data “speak”	impose physical laws
data driven	knowledge driven
adaptive models	differential equations
digit recognition	

General Approach

Broadly Speaking: Two approaches to modeling

data modeling	*mechanistic modeling*
let the data "speak"	impose physical laws
data driven	knowledge driven
adaptive models	differential equations
digit recognition	climate, weather models

Dimensionality Reduction

- Linear relationship between the data, $\mathbf{Y} \in \Re^{N \times D}$, and a reduced dimensional representation, $\mathbf{U} \in \Re^{N \times q}$, where $q \ll D$.

$$\mathbf{Y} = \mathbf{U}\mathbf{W} + \boldsymbol{\epsilon},$$

$$\boldsymbol{\epsilon} \sim \mathcal{N}\left(\mathbf{0}, \boldsymbol{\Sigma}\right)$$

- Integrate out $\mathbf{U}$, optimize with respect to $\mathbf{W}$.
- For Gaussian prior, $\mathbf{U} \sim \mathcal{N}\left(\mathbf{0}, \mathbf{I}\right)$
  - and $\Sigma = \sigma^2\mathbf{I}$ we have probabilistic PCA (Tipping and Bishop, 1999; Roweis, 1998).
  - and Σ constrained to be diagonal, we have factor analysis.

Dimensionality Reduction: Temporal Data

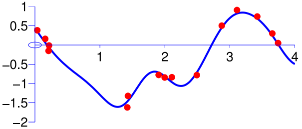
- Deal with temporal data with a temporal latent prior.
- Independent Gauss-Markov priors over each $u_i(t)$ leads to : Rauch-Tung-Striebel (RTS) smoother (Kalman filter).
- More generally consider a Gaussian process (GP) prior,

$$p\left(\mathbf{U}|\mathbf{t}\right) = \prod_{i=1}^{q} \mathcal{N}\left(\mathbf{u}_{:,i}|\mathbf{0}, \mathbf{K}_{u_{:,i},u_{:,i}}\right).$$

- Given the covariance functions for $\{u_i(t)\}$ we have an implied covariance function across all $\{y_i(t)\}$—(ML: semi-parametric latent factor model (Teh et al., 2005), Geostatistics: linear model of coregionalization).
- Rauch-Tung-Striebel smoother has been preferred
  - linear computational complexity in N.
  - Advances in sparse approximations have made the general GP framework practical. (Titsias, 2009; Snelson and Ghahramani, 2006; Quiñonero Candela and Rasmussen, 2005).

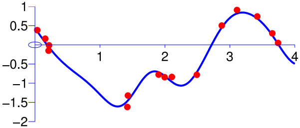
Outline

Introduction: covariances for multiple outputs



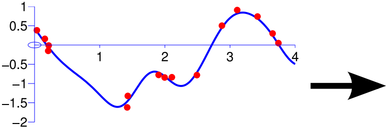
$$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$$

Introduction: covariances for multiple outputs



$$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$$

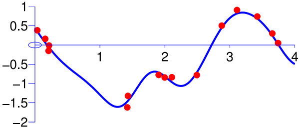
Introduction: covariances for multiple outputs



$k^1(\mathbf{x}_i, \mathbf{x}_j)$

$$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$$

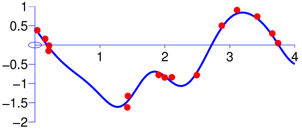
Introduction: covariances for multiple outputs



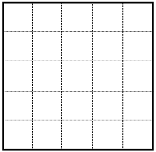
$$k^1(\mathbf{x}_i, \mathbf{x}_j)$$

$$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$$

Introduction: covariances for multiple outputs

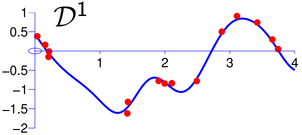


$\rightarrow k^1(\mathbf{x}_i, \mathbf{x}_j) \rightarrow \mathbf{K}^1 =$

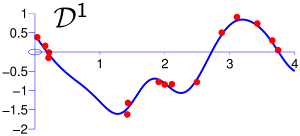


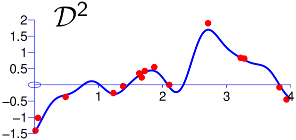
$$\mathcal{D}^1 = \{(\mathbf{x}_i^1, y_i^1) | i = 1, \ldots, N_1\}$$

Introduction: covariances for multiple outputs

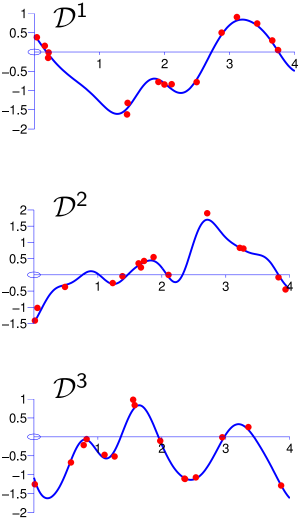


Introduction: covariances for multiple outputs

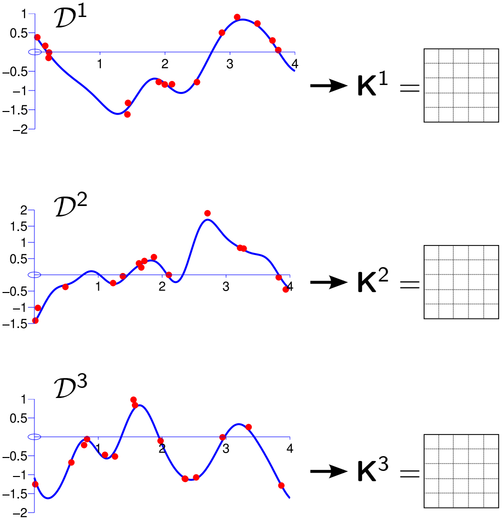




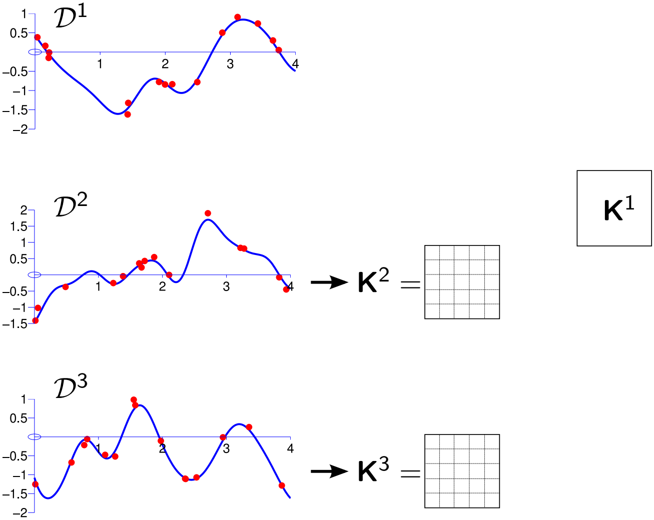
Introduction: covariances for multiple outputs



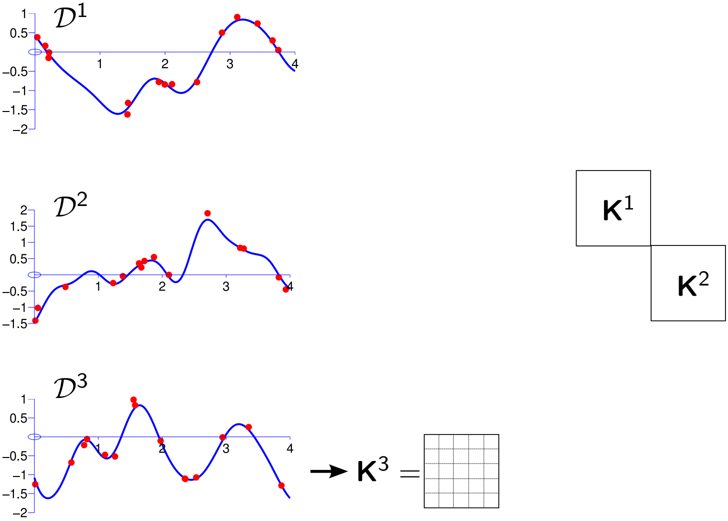
Introduction: covariances for multiple outputs



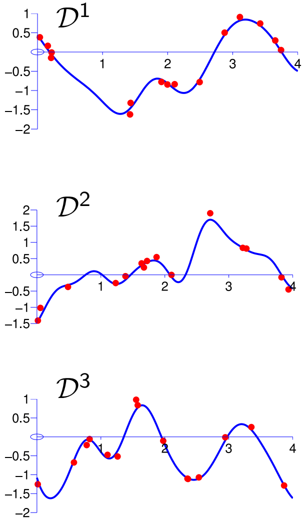
Introduction: covariances for multiple outputs

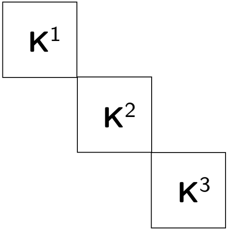


Introduction: covariances for multiple outputs

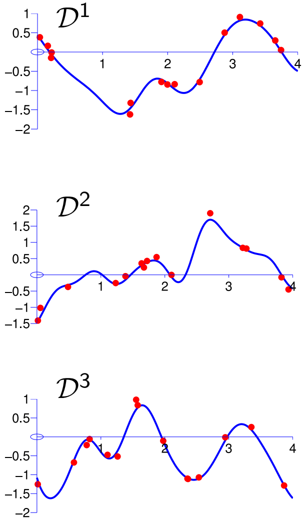


Introduction: covariances for multiple outputs



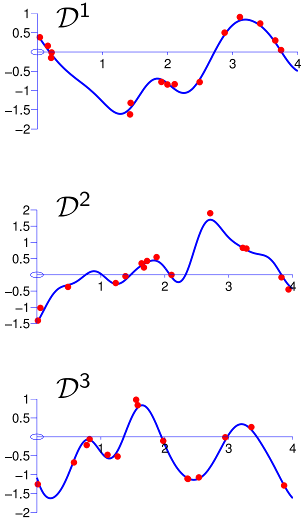


Introduction: covariances for multiple outputs



$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^1 & & \\ & \mathbf{K}^2 & \\ & & \mathbf{K}^3 \end{bmatrix}$$

Introduction: covariances for multiple outputs



Joint covariance

$$\mathbf{K} = \begin{bmatrix} \mathbf{K}^1 & ? & ? \\ ? & \mathbf{K}^2 & ? \\ ? & ? & \mathbf{K}^3 \end{bmatrix}$$

$\mathbf{K}$ be a valid covariance matrix

Some approaches

- Linear model of coregionalization.
- Intrinsic coregionalization model.
- Multitask kernels.
- Convolution of covariances.
- Convolution of processes or convolution process.

Convolution Process

- A convolution process is a moving-average construction that guarantees a valid covariance function.
- Consider a set of functions $\{f_j(\mathbf{x})\}_{j=1}^{D}$.
- Each function can be expressed as

$$f_j(\mathbf{x}) = \int_{\mathcal{X}} G_j(\mathbf{x}-\mathbf{z})u(\mathbf{z})\mathrm{d}\mathbf{z} = G_j(\mathbf{x}) * u(\mathbf{x}).$$

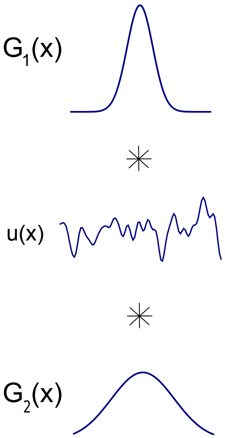
- Influence of more than one latent function, $\{u_i(\mathbf{z})\}_{i=1}^{q}$ and inclusion of an independent process $w_j(\mathbf{x})$

$$y_j(\mathbf{x}) = f_j(\mathbf{x}) + w_j(\mathbf{x}) = \sum_{i=1}^{q} \int_{\mathcal{X}} G_{j,i}(\mathbf{x}-\mathbf{z})u_i(\mathbf{z})\mathrm{d}\mathbf{z} + w_j(\mathbf{x}).$$

A pictorial representation

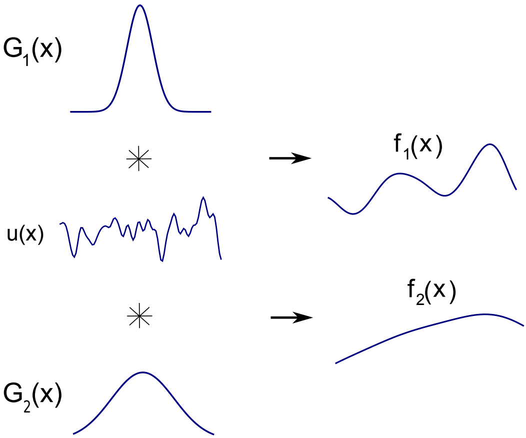
u(x): latent function.

A pictorial representation



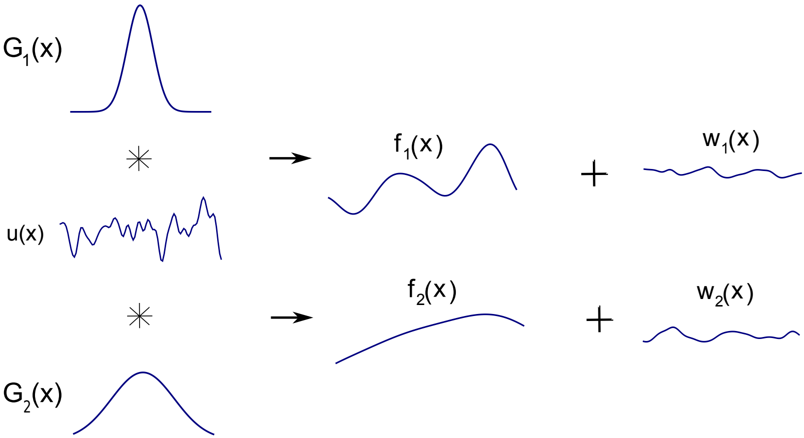
u(x): latent function.
G(x): smoothing kernel.

A pictorial representation



u(x): latent function.
G(x): smoothing kernel.
f(x): output function.

A pictorial representation



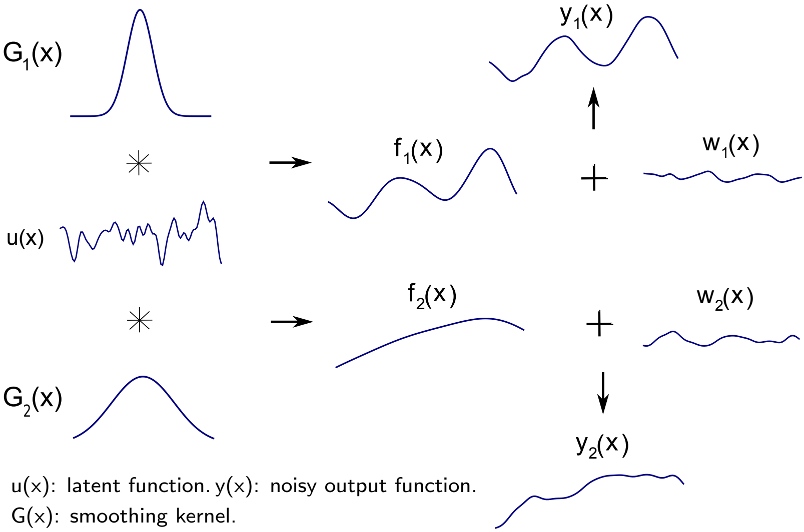
u(x): latent function.

G(x): smoothing kernel.

f(x): output function.

w(x): independent process.

A pictorial representation



u(x): latent function. y(x): noisy output function.

G(x): smoothing kernel.

f(x): output function.

w(x): independent process.

Covariance of the output functions.

The covariance between $y_j(\mathbf{x})$ and $y_{j'}(\mathbf{x}')$ is given as

$$\mathrm{cov}\left[y_j(\mathbf{x}), y_{j'}(\mathbf{x}')\right] = \mathrm{cov}\left[f_j(\mathbf{x}), f_{j'}(\mathbf{x}')\right] + \mathrm{cov}\left[w_j(\mathbf{x}), w_{j'}(\mathbf{x}')\right]\delta_{j,j'}$$

where

$$\mathrm{cov}\left[f_j(\mathbf{x}), f_{j'}(\mathbf{x}')\right] = \int_{\mathcal{X}} G_j(\mathbf{x}-\mathbf{z}) \int_{\mathcal{X}} G_{j'}(\mathbf{x}'-\mathbf{z}')\mathrm{cov}\left[u(\mathbf{z}), u(\mathbf{z}')\right] \mathrm{d}\mathbf{z}'\mathrm{d}\mathbf{z}$$

Different forms of covariance for the output functions.

- Input *Gaussian process*

$$\operatorname{cov}\left[f_j, f_{j'}\right] = \int_{\mathcal{X}} G_j(\mathbf{x}-\mathbf{z}) \int_{\mathcal{X}} G_{j'}(\mathbf{x}'-\mathbf{z}') k_{u,u}(\mathbf{z},\mathbf{z}') \mathrm{d}\mathbf{z}' \mathrm{d}\mathbf{z}$$

- Input *white noise process*

$$\operatorname{cov}\left[f_j, f_{j'}\right] = \int_{\mathcal{X}} G_j(\mathbf{x}-\mathbf{z}) G_{j'}(\mathbf{x}'-\mathbf{z}) \mathrm{d}\mathbf{z}$$

- Covariance between output functions and latent functions

$$\operatorname{cov}\left[f_j, u\right] = \int_{\mathcal{X}} G_j(\mathbf{x}-\mathbf{z}') k_{u,u}(\mathbf{z}',\mathbf{z}) \mathrm{d}\mathbf{z}'$$

Likelihood of the full Gaussian process.

- The likelihood of the model is given by

$$p(\mathbf{y}|\mathbf{X}, \phi) = \mathcal{N}(\mathbf{0}, \mathbf{K}_{f,f} + \mathbf{\Sigma})$$

  where $\mathbf{y} = \left[\mathbf{y}_1^\top, \ldots, \mathbf{y}_D^\top\right]^\top$ is the set of output functions, $\mathbf{K}_{f,f}$ covariance matrix with blocks $\mathrm{cov}\left[f_j, f_{j'}\right]$, $\mathbf{\Sigma}$ matrix of noise variances, ϕ is the set of parameters of the covariance matrix and $\mathbf{X} = \{\mathbf{x}_1, \ldots, \mathbf{x}_N\}$ is the set of input vectors.

- Learning from the log-likelihood involves the inverse of $\mathbf{K}_{f,f} + \mathbf{\Sigma}$, which grows with complexity $\mathcal{O}(N^3D^3)$

Predictive distribution of the full Gaussian process.

- Predictive distribution at $\mathbf{X}_*$

$$p(\mathbf{y}_*|\mathbf{y}, \mathbf{X}, \mathbf{X}_*, \phi) = \mathcal{N}(\boldsymbol{\mu}_*, \boldsymbol{\Lambda}_*)$$

with

$$\boldsymbol{\mu}_* = \mathbf{K}_{f_*,f}(\mathbf{K}_{f,f} + \boldsymbol{\Sigma})^{-1}\mathbf{y}$$
$$\boldsymbol{\Lambda}_* = \mathbf{K}_{f_*,f_*} - \mathbf{K}_{f_*,f}(\mathbf{K}_{f,f} + \boldsymbol{\Sigma})^{-1}\mathbf{K}_{f,f_*} + \boldsymbol{\Sigma}$$

- Prediction is $\mathcal{O}(DN)$ for the mean and $\mathcal{O}(D^2N^2)$ for the variance, for one test point. Storage is $\mathcal{O}(D^2N^2)$.

Mechanical Analogy

Back to Latent Force Models!

- These models rely on the latent variables to provide the dynamic information.
- We now introduce a further dynamical system with a *mechanistic* inspiration.
- Physical Interpretation:
  - the latent functions, $u_i(t)$ are q forces.
  - We observe the displacement of D springs to the forces.,
  - Interpret system as the force balance equation, $\mathbf{Y}\mathbf{D} = \mathbf{U}\mathbf{S} + \epsilon$.
  - Forces act, e.g. through levers — a matrix of sensitivities, $\mathbf{S} \in \Re^{q \times D}$.
  - Diagonal matrix of spring constants, $\mathbf{D} \in \Re^{D \times D}$.
  - Original System: $\mathbf{W} = \mathbf{S}\mathbf{D}^{-1}$.

Extend Model

- Add a damper and give the system mass.

$$\mathbf{US} = \ddot{\mathbf{Y}}\mathbf{M} + \dot{\mathbf{Y}}\mathbf{C} + \mathbf{Y}\mathbf{D} + \epsilon.$$

- Now have a second order mechanical system.
- It will exhibit inertia and resonance.
- There are many systems that can also be represented by differential equations.
  - When being forced by latent function(s), $\{u_i(t)\}_{i=1}^{q}$, we call this a *latent force model*.

Outline

- For Gaussian process we can compute the covariance matrices for the output displacements.
- For one displacement the model is

$$m_k \ddot{y}_k(t) + c_k \dot{y}_k(t) + d_k y_k(t) = b_k + \sum_{i=0}^{q} s_{ik} u_i(t), \qquad (1)$$

  where, m_k is the kth diagonal element from $\mathbf{M}$ and similarly for c_k and d_k. s_{ik} is the i, kth element of $\mathbf{S}$.
- Model the latent forces as q independent, GPs with RBF covariances

$$k_{u_i u_l}(t, t') = \exp\left(-\frac{(t - t')^2}{\ell_i^2}\right) \delta_{il}.$$

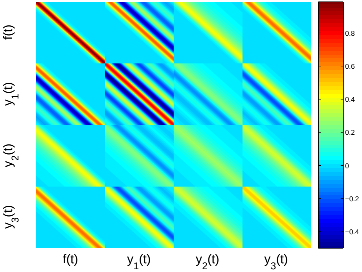
Covariance for ODE Model

- RBF Kernel function for $u(t)$

$$y_j(t) = \frac{1}{m_j\omega_j}\sum_{i=1}^{q} s_{ji}\exp(-\alpha_j t)\int_0^t u_i(\tau)\exp(\alpha_j\tau)\sin(\omega_j(t-\tau))\mathrm{d}\tau$$

- Joint distribution for $y_1(t)$, $y_2(t)$, $y_3(t)$ and $u(t)$.
  Damping ratios:

ζ_1	ζ_2	ζ_3
0.125	2	1



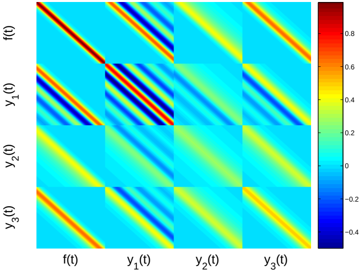
Covariance for ODE Model

- Analogy

$$y = \sum_i \mathbf{e}_i^\top \mathbf{u}_i \quad \mathbf{u}_i \sim \mathcal{N}\left(\mathbf{0}, \Sigma_i\right) \rightarrow y \sim \mathcal{N}\left(0, \sum_i \mathbf{e}_i^\top \Sigma_i \mathbf{e}_i\right)$$

- Joint distribution for $y_1(t)$, $y_2(t)$, $y_3(t)$ and $u(t)$. Damping ratios:

ζ_1	ζ_2	ζ_3
0.125	2	1



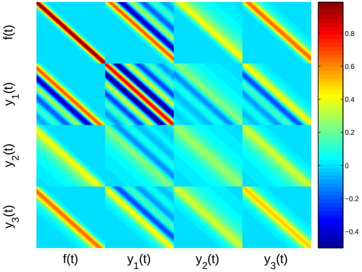
Covariance for ODE Model

- RBF Kernel function for $u(t)$

$$y_j(t) = \frac{1}{m_j \omega_j} \sum_{i=1}^{q} s_{ji} \exp(-\alpha_j t) \int_0^t u_i(\tau) \exp(\alpha_j \tau) \sin(\omega_j (t - \tau)) \mathrm{d}\tau$$

- Joint distribution for $y_1(t)$, $y_2(t)$, $y_3(t)$ and $u(t)$. Damping ratios:

ζ_1	ζ_2	ζ_3
0.125	2	1



Joint Sampling of $y(t)$ and $u(t)$

- `lfmSample`

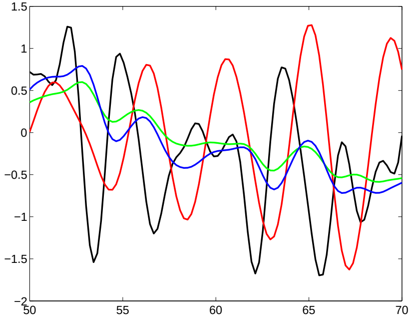


Figure: Joint samples from the ODE covariance, *black*: $u(t)$, *red*: $y_1(t)$ (underdamped), *green*: $y_2(t)$ (overdamped), and *blue*: $y_3(t)$ (critically damped).

Joint Sampling of $y(t)$ and $u(t)$

- `lfmSample`

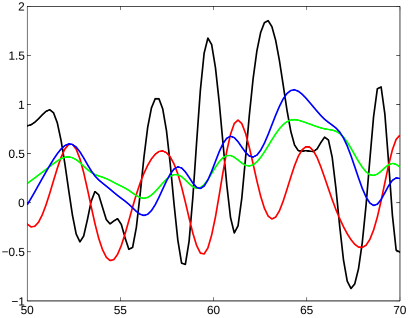


Figure: Joint samples from the ODE covariance, *black*: $u(t)$, *red*: $y_1(t)$ (underdamped), *green*: $y_2(t)$ (overdamped), and *blue*: $y_3(t)$ (critically damped).

Joint Sampling of $y(t)$ and $u(t)$

- `lfmSample`

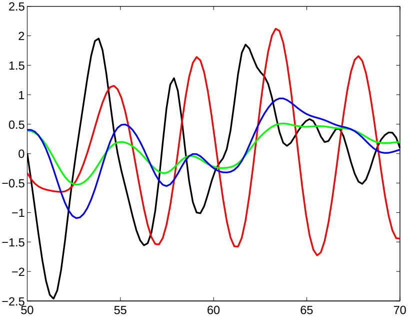


Figure: Joint samples from the ODE covariance, *black*: $u(t)$, *red*: $y_1(t)$ (underdamped), *green*: $y_2(t)$ (overdamped), and *blue*: $y_3(t)$ (critically damped).

Joint Sampling of $y(t)$ and $u(t)$

- `lfmSample`

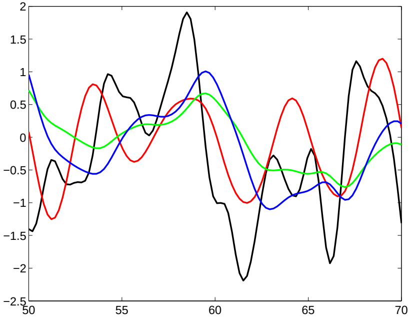


Figure: Joint samples from the ODE covariance, *black*: $u(t)$, *red*: $y_1(t)$ (underdamped), *green*: $y_2(t)$ (overdamped), and *blue*: $y_3(t)$ (critically damped).

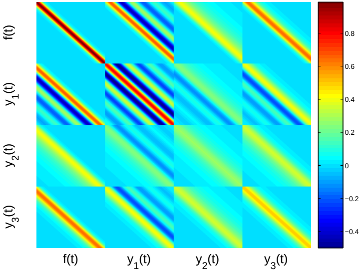
Covariance for ODE

- RBF Kernel function for $u(t)$

$$y_j(t) = \frac{1}{m_j\omega_j}\sum_{i=1}^{q} s_{ji}\exp(-\alpha_j t)\int_0^t u_i(\tau)\exp(\alpha_j\tau)\sin(\omega_j(t-\tau))\mathrm{d}\tau$$

- Joint distribution for $y_1(t)$, $y_2(t)$, $y_3(t)$ and $u(t)$.
- Damping ratios:

ζ_1	ζ_2	ζ_3
0.125	2	1



Outline

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Example: Motion Capture

Mauricio Alvarez and David Luengo (Álvarez et al., 2009)

- Motion capture data: used for animating human motion.
- Multivariate time series of angles representing joint positions.
- Objective: generalize from training data to realistic motions.
- Use 2nd Order Latent Force Model with mass/spring/damper (resistor inductor capacitor) at each joint.

Prediction of Test Motion

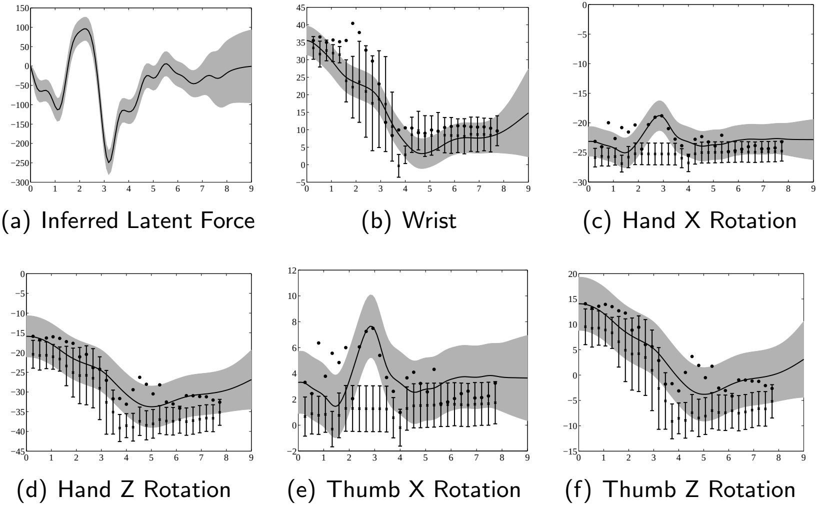
- Model left arm only.
- 3 balancing motions (18, 19, 20) from subject 49.
- 18 and 19 are similar, 20 contains more dramatic movements.
- Train on 18 and 19 and testing on 20
- Data was down-sampled by 32 (from 120 fps).
- Reconstruct motion of left arm for 20 given other movements.
- Compare with GP that predicts left arm angles given other body angles.

Mocap Results

Table: Root mean squared (RMS) angle error for prediction of the left arm's configuration in the motion capture data. Prediction with the latent force model outperforms the prediction with regression for all apart from the radius's angle.

Angle	Latent Force Error	Regression Error
Radius	4.11	4.02
Wrist	6.55	6.65
Hand X rotation	1.82	3.21
Hand Z rotation	2.76	6.14
Thumb X rotation	1.77	3.10
Thumb Z rotation	2.73	6.09

Mocap Results II



(a) Inferred Latent Force (b) Wrist (c) Hand X Rotation

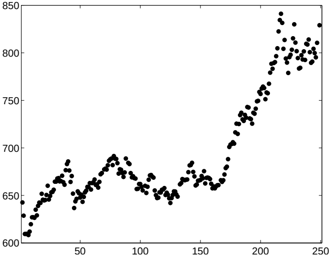
(d) Hand Z Rotation (e) Thumb X Rotation (f) Thumb Z Rotation

Figure: Predictions from LFM (solid line, grey error bars) and direct regression (crosses with stick error bars).

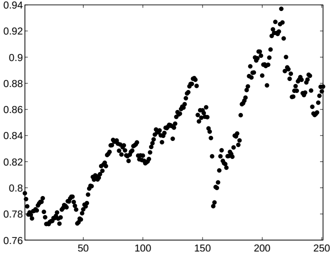
Outline

Financial data set

Multivariate financial data set: the dollar prices of the 3 precious metals and top 10 currencies.



(a) Gold



(b) AUD

Dynamic model

- Our model: a set of coupled differential equations, driven by either a smooth Gaussian process, a white noise process, or both,

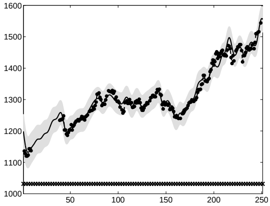
$$\frac{\mathrm{d}f_j(t)}{\mathrm{d}t} = \sum_{i=1}^{q} s_{i,j} u_i(t) - d_j f_j(t),$$

  where d_j is a decay coefficient and $s_{i,j}$ quantifies the influence of the process $u_i(t)$.

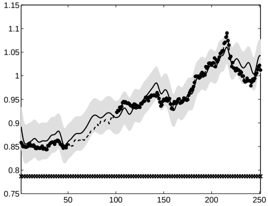
- If $\{u_i(t)\}_{i=1}^{q}$ are white noise processes $\rightarrow$ Langevin equation $\rightarrow$ a linear stochastic differential equation.

- Solution for $f_j(t)$ has the form of convolutions. For a single output and white noise process, $f_j(t) \rightarrow$ Ornstein-Uhlenbeck (OU) process.

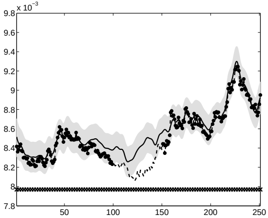
Some results



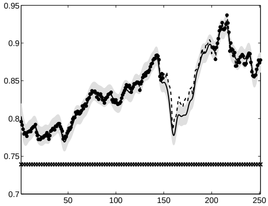
(c) XPT: Real data and prediction



(d) CAD: Real data and prediction



(e) JPY: Real data and prediction



(f) AUD: Real data and prediction

Outline

Discussion and Future Work

- Integration of probabilistic inference with mechanistic models.
- Ongoing/other work:
  - Non linear response and non linear differential equations.
  - Scaling up to larger systems
  - Robotics applications
  - Computational biology applications
  - Applications to other types of system, *e.g.* spatial systems.

Acknowledgements

Funding **Google Research Award**: "Mechanistically Inspired Convolution Processes for Learning". David Luengo's visit to Manchester was financed by the Comunidad de Madrid (project PRO-MULTIDIS-CM, S-0505/TIC/0233), and by the Spanish government (CICYT project TEC2006-13514-C02-01 and researh grant JC2008-00219).

References I

M. Álvarez, D. Luengo, and N. D. Lawrence. Latent force models. In van Dyk and Welling (2009), pages 9–16. [PDF].

J. Quiñonero Candela and C. E. Rasmussen. A unifying view of sparse approximate Gaussian process regression. *Journal of Machine Learning Research*, 6:1939–1959, 2005.

S. T. Roweis. EM algorithms for PCA and SPCA. In M. I. Jordan, M. J. Kearns, and S. A. Solla, editors, *Advances in Neural Information Processing Systems*, volume 10, pages 626–632, Cambridge, MA, 1998. MIT Press.

E. Snelson and Z. Ghahramani. Sparse Gaussian processes using pseudo-inputs. In Y. Weiss, B. Schölkopf, and J. C. Platt, editors, *Advances in Neural Information Processing Systems*, volume 18, Cambridge, MA, 2006. MIT Press.

Y. W. Teh, M. Seeger, and M. I. Jordan. Semiparametric latent factor models. In R. G. Cowell and Z. Ghahramani, editors, *Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics*, pages 333–340, Barbados, 6-8 January 2005. Society for Artificial Intelligence and Statistics.

M. E. Tipping and C. M. Bishop. Probabilistic principal component analysis. *Journal of the Royal Statistical Society, B*, 6(3):611–622, 1999. [PDF]. [DOI].

M. K. Titsias. Variational learning of inducing variables in sparse Gaussian processes. In van Dyk and Welling (2009), pages 567–574.

D. van Dyk and M. Welling, editors. *Artificial Intelligence and Statistics*, volume 5, Clearwater Beach, FL, 16-18 April 2009. JMLR W&CP 5.