Multi-task Gaussian Process Prediction

Chris Williams
Joint Work with Edwin Bonilla, Kian Ming A. Chai,
Stefan Klanke and Sethu Vijayakumar

Institute for Adaptive and Neural Computation
School of Informatics
University of Edinburgh, UK

September 2008

Motivation: Multi-task Learning

- Sharing information across tasks
- e.g. Exam score prediction, compiler performance prediction, robot inverse dynamics

Motivation: Multi-task Learning

- Sharing information across tasks
- e.g. Exam score prediction, compiler performance prediction, robot inverse dynamics

- Assuming task relatedness can be detrimental (Caruana, 1997; Baxter, 2000)
- Task descriptors unavailable or difficult to define
  - e.g. Compiler performance prediction: code features, responses

Motivation: Multi-task Learning

- Sharing information across tasks
- e.g. Exam score prediction, compiler performance prediction, robot inverse dynamics

- Assuming task relatedness can be detrimental (Caruana, 1997; Baxter, 2000)
- Task descriptors unavailable or difficult to define
  - e.g. Compiler performance prediction: code features, responses

- Learning inter-task dependencies based on task identities
- Correlations between tasks directly induced
- GP framework

Outline

- The Model
- Making Predictions and Learning Hyperparameters
- Cancellation of Transfer
- Related Work
- Experiments and Results
- MTL in Robot Inverse Dynamics
- Conclusions and Discussion

Multi-task Setting

Given a set X of N distinct inputs $\mathbf{x}_1, \ldots, \mathbf{x}_N$:

- Complete set of responses:

$$\mathbf{y} = (y_{11}, \ldots, y_{N1}, \ldots, y_{12}, \ldots, y_{N2}, \ldots, y_{1M}, \ldots, y_{NM})^{\mathrm{T}}$$

$y_{i\ell}$: response for the ℓ^{th} task on the i^{th} input $\mathbf{x}_i$

Y: $N \times M$ matrix such $\mathbf{y} = \text{vec}\, Y$

- Goal: Given observations $\mathbf{y}_o \subset \mathbf{y}$:
  - make predictions of unobserved values $\mathbf{y}_u$

Multi-task GP

We place a (zero mean) GP prior over the latent functions $\{f_\ell\}$:

The Model

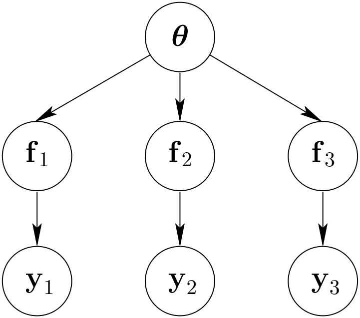
$$\langle f_\ell(\mathbf{x}) f_m(\mathbf{x}')\rangle = K^f_{\ell m} k^x(\mathbf{x}, \mathbf{x}') \qquad y_{i\ell} \sim \mathcal{N}(f_\ell(\mathbf{x}_i), \sigma^2_\ell),$$

- K^f: PSD matrix that specifies the inter-task similarities
- k^x: Covariance function over inputs
- σ^2_ℓ: Noise variance for the ℓ^{th} task.

Additionally, k^x:

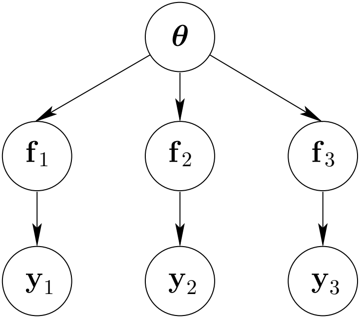
- stationary, *correlation* function
- e.g. squared exponential

Multi-task GP (2)

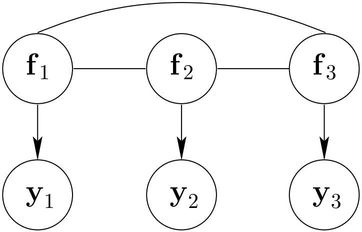


Other approaches

Multi-task GP (2)

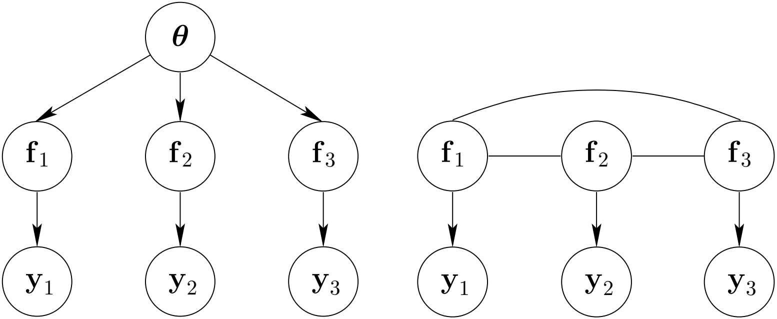


Other approaches



Our approach

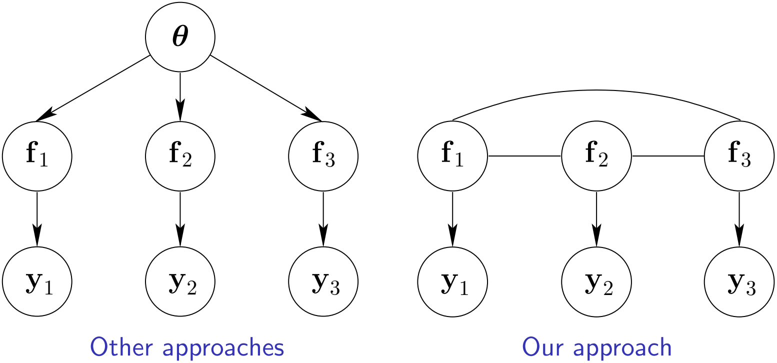
Multi-task GP (2)



Other approaches Our approach

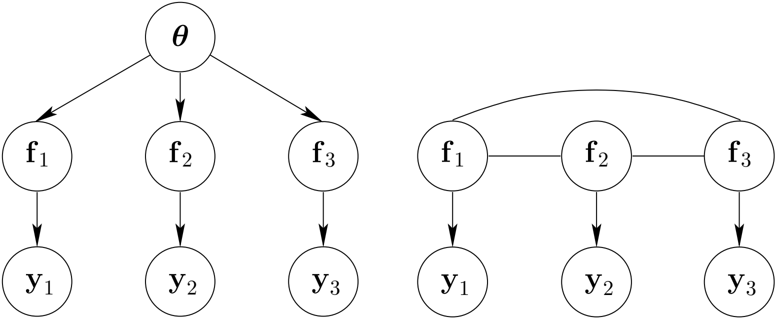
- Observations on one task can affect predictions on the others

Multi-task GP (2)



- Observations on one task can affect predictions on the others
- Bonilla et. al (2007), Yu et. al (2007): $K^f_{\ell m} = k^f(\mathbf{t}_\ell, \mathbf{t}_m)$

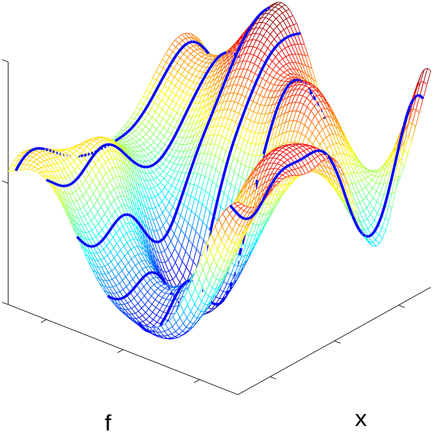
Multi-task GP (2)



Other approaches

Our approach

- Observations on one task can affect predictions on the others
- Bonilla et. al (2007), Yu et. al (2007): $K^f_{\ell m} = k^f(\mathbf{t}_\ell, \mathbf{t}_m)$
- Multi-task clustering easily modelled


f
x

Making Predictions

The mean prediction on a new data-point $\mathbf{x}_*$ for task ℓ is given by:

$$\begin{aligned} \bar{f}_\ell(\mathbf{x}_*) &= (\mathbf{k}^f_\ell \otimes \mathbf{k}^x_*)^T \Sigma^{-1} \mathbf{y}, \text{ with} \\ \Sigma &= K^f \otimes K^x + D \otimes I \end{aligned}$$

where:

- $\mathbf{k}^f_\ell$ selects the ℓ^{th} column of K^f
- $\mathbf{k}^x_*$: vector of covariances between $\mathbf{x}_*$ and the training points
- K^x: matrix of covariances between all pairs of training points
- D: diagonal matrix in which the $(\ell, \ell)^{\text{th}}$ element is σ^2_ℓ

Learning Hyperparameters

Given $\mathbf{y}_o$:

- Learn $\boldsymbol{\theta}_x$ of k^x, K^f, σ_ℓ^2 to maximize $p(\mathbf{y}_o|X)$.
- We note that: $\mathbf{y}|X \sim \mathcal{N}(\mathbf{0}, \Sigma)$

(a) Gradient-based method:

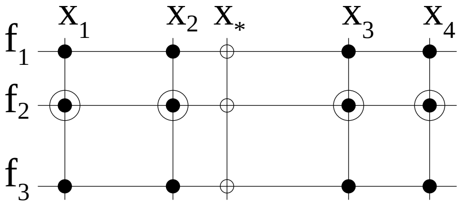
- $K^f = LL^T$ (Recall K^f must be PSD)
- Kronecker structure

(b) EM:

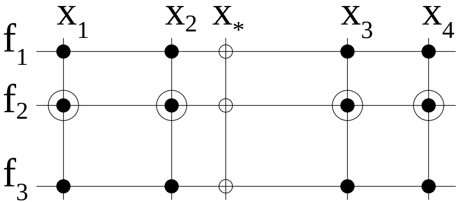
- learning of $\boldsymbol{\theta}_x$ and K^f in the M-step is decoupled
- closed-form updates for K^f and D
- K^f guaranteed PSD

$$\widehat{K}^f = N^{-1} \left\langle F^{\mathrm{T}} \left(K^x(\widehat{\boldsymbol{\theta}_x}) \right)^{-1} F \right\rangle$$

Noiseless observations + grid = Cancellation of Transfer



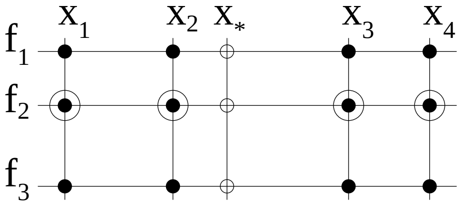
Noiseless observations + grid = Cancellation of Transfer



We can show that *if there is a grid design* and *no observation noise* then:

$$\overline{f}(\mathbf{x}_*, \ell) = (\mathbf{k}_*^x)^{\mathrm{T}}(K^x)^{-1}\mathbf{y}_{\cdot\ell}$$

Noiseless observations + grid = Cancellation of Transfer



We can show that *if there is a grid design* and *no observation noise* then:

$$\bar{f}(\mathbf{x}_*, \ell) = (\mathbf{k}_*^x)^{\mathrm{T}} (K^x)^{-1} \mathbf{y}_{\cdot\ell}$$

- The predictions for task ℓ depend only on the targets $\mathbf{y}_{\cdot\ell}$
- Similar result for the covariances
- This is know as *autokrigeability* in geostatistics

Related Work

- Early work on MTL: Thrun (1996), Caruana (1997)
- Minka (1997) and some other later GP work assumes that multiple tasks share the same hyperparameters but are otherwise uncorrelated
- Co-kriging in geostatistics
- Evgeniou et al (2005) induce correlations between tasks based on a correlated prior over linear regression parameters
- Conti & O'Hagan (2007): emulating multi-output simulators
- Use of task descriptors so that $K^f_{\ell m} = k^f(\mathbf{t}_\ell, \mathbf{t}_m)$, e.g. Yu et al (2007), Bonilla et al (2007).
- Semiparametric latent factor model (SLFM) of Teh et al (2005) has P latent processes each with its own covariance function. Noiseless outputs are obtained by linear mixing of these latent functions.
- Our model is similar, but simpler, in that all of the P latent processes share the same covariance function; this reduces the number of free parameters to be fitted and should help to minimize overfitting

Experiments

Compiler performance prediction

- y: *Speed-up* of a program (task) when applying a transformation sequence $\mathbf{x}$
- 11 C programs, 13 transformations, 5-length sequences
- "bag-of-characters" representation for $\mathbf{x}$

Experiments

Compiler performance prediction

- y: *Speed-up* of a program (task) when applying a transformation sequence $\mathbf{x}$
- 11 C programs, 13 transformations, 5-length sequences
- "bag-of-characters" representation for $\mathbf{x}$

Exam score prediction

- y: *Exam score* obtained by a student $\mathbf{x}$ in a specific school (task).
- 139 schools, 15362 students
- Student features ($\mathbf{x}$): exam year, gender, VR band, ethnic group
- dummy variables created

Results: School Data

- 10 random splits of the data into training (75%) and test (25%)
- k^x is squared exponential kernel, $K^f = LL^T$ with rank constraints
- % of variance explained (larger figures are better):

no transfer	task-descriptor	rank 1	rank 2	rank 3	rank 5
21.05 (1.15)	31.57 (1.61)	27.02 (2.03)	29.20 (1.60)	24.88 (1.62)	21.00 (2.42)

Results: School Data

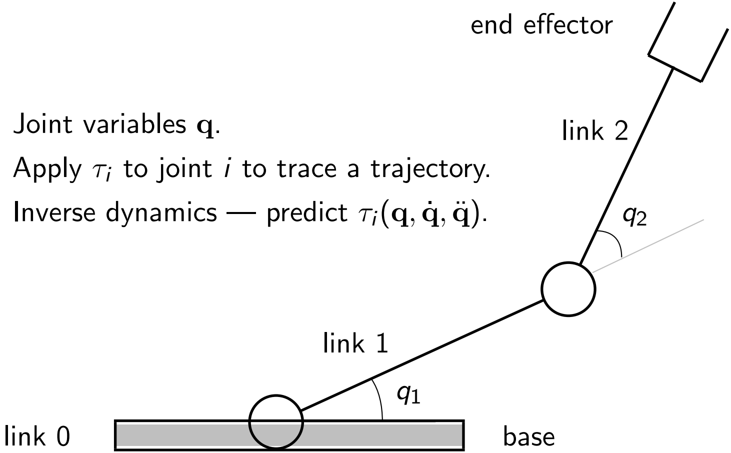
- 10 random splits of the data into training (75%) and test (25%)
- k^x is squared exponential kernel, $K^f = LL^T$ with rank constraints
- % of variance explained (larger figures are better):

no transfer	task-descriptor	rank 1	rank 2	rank 3	rank 5
21.05 (1.15)	31.57 (1.61)	27.02 (2.03)	29.20 (1.60)	24.88 (1.62)	21.00 (2.42)

- Better results with multi-task learning than without
- Task-descriptor approach slightly outperforms "free-form" method

Multi-task Learning in Robot Inverse Dynamics

- Joint variables $\mathbf{q}$.
- Apply τ_i to joint i to trace a trajectory.
- Inverse dynamics — predict $\tau_i(\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$.



Inverse Dynamics
Characteristics of $\boldsymbol{\tau}$

- Torques are non-linear functions of $\mathbf{x} \stackrel{\text{def}}{=} (\mathbf{q}, \dot{\mathbf{q}}, \ddot{\mathbf{q}})$.
- (One) idealized rigid body control:

$$\tau_i(\mathbf{x}) = \underbrace{\mathbf{b}_i^{\mathrm{T}}(\mathbf{q})\ddot{\mathbf{q}} + \dot{\mathbf{q}}^{\mathrm{T}} H_i(\mathbf{q})\dot{\mathbf{q}}}_{\text{kinetic}} + \overbrace{g_i(\mathbf{q})}^{\text{potential}} + \underbrace{f_i^{\mathsf{v}}\dot{q}_i + f_i^{\mathsf{c}}\operatorname{sgn}(\dot{q}_i)}_{\text{viscous and Coulomb frictions}},$$

- Physics-based modelling can be hard due to factors like unknown parameters, friction and contact forces, joint elasticity, making analytical predictions unfeasible
- This is particularly true for compliant, lightweight humanoid robots

Inverse Dynamics
Characteristics of $\boldsymbol{\tau}$

- Functions *change* with the loads handled at the end effector
- Loads have different mass, shapes, sizes.
- Bad news (1): Need a different inverse dynamics model for different loads.
- Bad news (2): Different loads may go through different trajectory in data collection phase and may explore different portions of the $\mathbf{x}$-space.

- Good news: the changes enter through changes in the dynamic parameters of the last link
- Good news: changes are linear wrt the dynamic parameters

$$\tau_i^m(\mathbf{x}) = \mathbf{y}_i^T(\mathbf{x})\boldsymbol{\pi}^m$$

  where $\boldsymbol{\pi}^m \in \mathbb{R}^{11}$ (e.g. Petkos and Vijayakumar,2007)
- Reparameterization:

$$\tau_i^m(\mathbf{x}) = \mathbf{y}_i^T(\mathbf{x})\boldsymbol{\pi}^m = \mathbf{y}_i^T(\mathbf{x})A_i^{-1}A_i\boldsymbol{\pi}^m = \mathbf{z}_i^T(\mathbf{x})\boldsymbol{\rho}_i^m$$

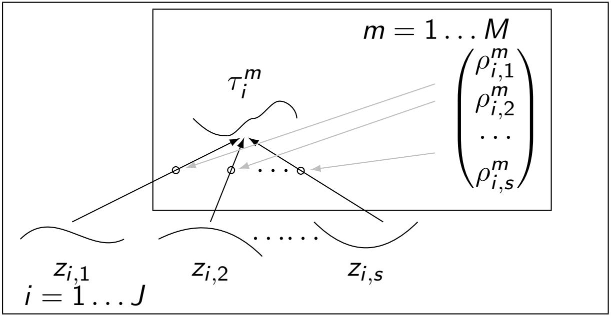
  where A is a non-singular 11×11 matrix

GP prior for Inverse Dynamics for multiple loads

- Independent GP priors over the functions $z_{ij}(\mathbf{x}) \Rightarrow$ multi-task GP prior over τ_i^ms

$$\left\langle \tau_i^\ell(\mathbf{x})\tau_i^m(\mathbf{x}') \right\rangle = (K_i^\rho)_{\ell m} k_i^x(\mathbf{x}, \mathbf{x}')$$

- $K_i^\rho \in \mathbb{R}^{M\times M}$ is a task (or context) similarity matrix with $(K_i^\rho)_{\ell m} = (\boldsymbol{\rho}_i^m)^T \boldsymbol{\rho}_i^\ell$



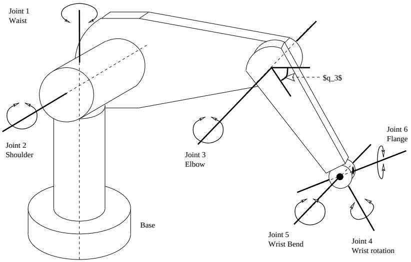
GP prior for $c(\mathbf{x}, \mathbf{x}')$

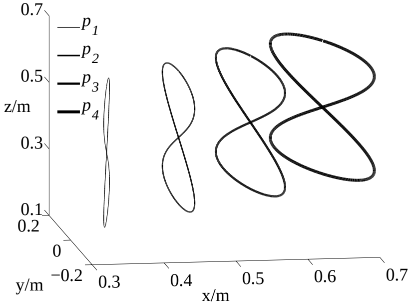
$$
\begin{aligned}
c(\mathbf{x}, \mathbf{x}') = \text{bias} &+ [\text{linear with ARD}](\mathbf{x}, \mathbf{x}') \\
&+ [\text{squared exponential with ARD}](\mathbf{x}, \mathbf{x}') \\
&+ [\text{linear (with ARD)}](\operatorname{sgn}(\dot{q}), \operatorname{sgn}(\dot{q}'))
\end{aligned}
$$

- Domain knowledge relates to last term (Coulomb friction)

Data

- Puma 560 robot arm manipulator: 6 degrees of freedom
- Realistic simulator (Corke, 1996), including viscous and asymmetric-Coulomb frictions.
- 4 paths $\times$ 4 speeds $=$ 16 different trajectories:
- Speeds: 5s, 10s, 15s and 20s completion times.
- 15 loads (contexts): 0.2kg . . . 3.0kg, various shapes and sizes.





Data

Training data

- 1 reference trajectory common to handling of all loads.
- 14 unique training trajectories, one for each context (load)
- 1 trajectory has no data for any context; thus this is always novel

Test data

- Interpolation data sets for testing on reference trajectory and the unique trajectory for each load.
- Extrapolation data sets for testing on all trajectories.

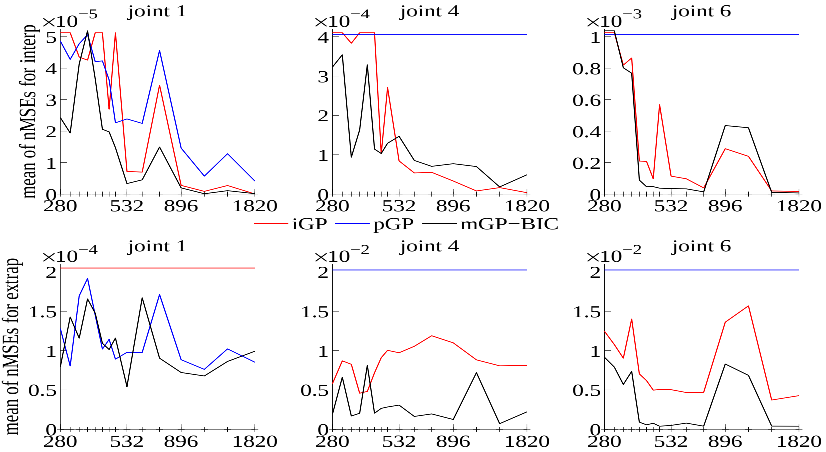
Methods

iGP	Independent GP	GPs trained independently for each load but tying parameters across loads
pGP	pooled GP	one single GP trained by pooling data across loads
mGP	multi-task GP with BIC	sharing latent functions across loads, selecting similarity matrix using BIC

- For mGP, the rank of K^f is determined using BIC criterion

Results

xaxis: total number of training datapoints, yaxis: nMSE
top: interpolation, bottom: extrapolation



Conclusions and Discussion

- GP formulation of MTL with factorization $k^x(\mathbf{x}, \mathbf{x}')$ and K^f, and encoding of task similarity
- This model fits exactly for multi-context inverse dynamics
- Results show that MTL can be effective
- This is one model for MTL, but what about others, e.g. cov functions that don't factorize?