Learning GPs from Multiple Tasks

Kai Yu[1]
Joint work with **Volker Tresp**[1], and **Anton Schwaighofer**[2]

[1] Corporate Technology, Siemens, Munich

[2] Intelligent Data Analysis, Fraunhofer FIRST, Berlin

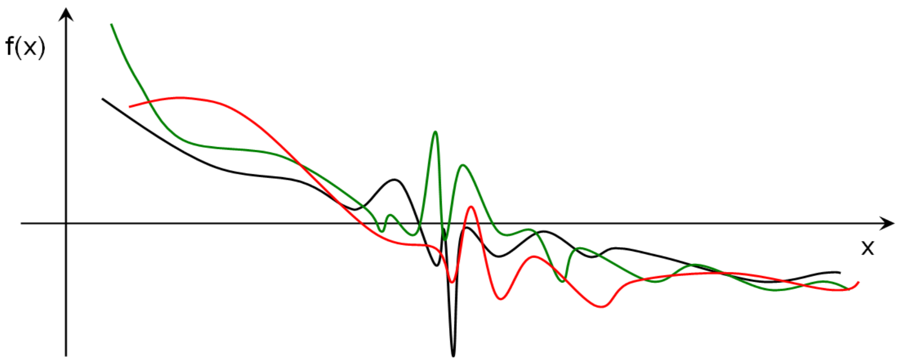
Some Real-world Problems

- **Text categorization**: One document belongs to multiple categories, which might be related semantically.
- **Preference prediction**: Users' interests mutually influence each other.
- **Computer vision**: The movements of different parts of a robot are mutually constrained.

Sometimes we have to model multiple dependent functions!

Modeling Dependency of Functions

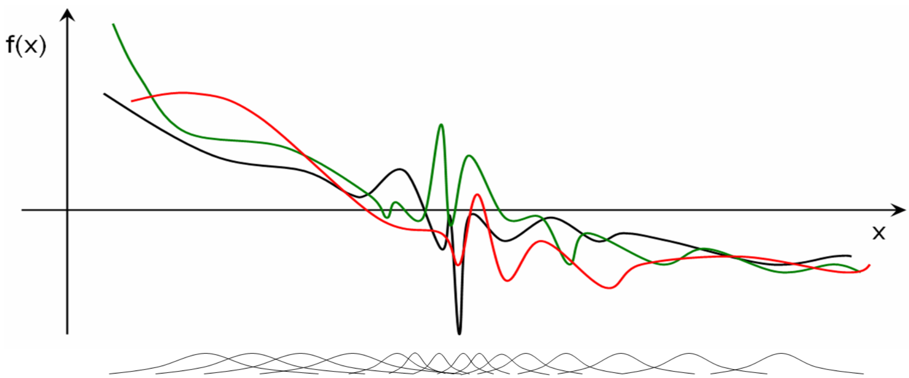
- Functions generated from **an unknown underlying process**



- **They share something in common**
  - **Mean** of those functions
  - **Local smoothness** of functions

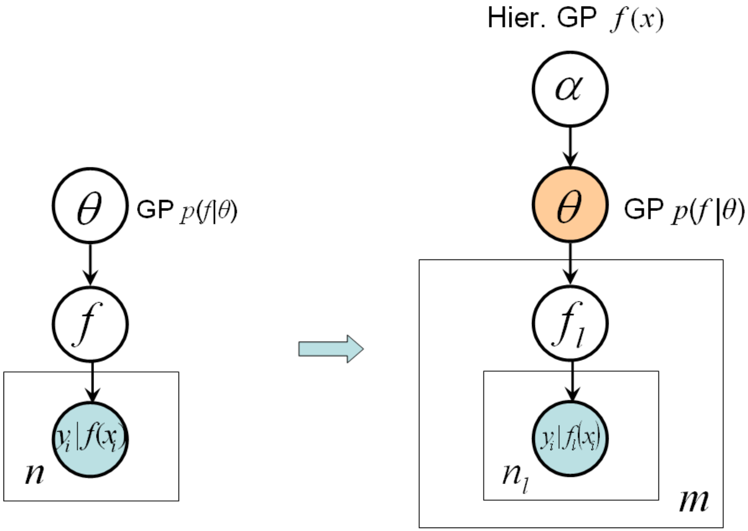
Modeling Dependency of Functions

- Functions generated from **an unknown underlying process**



- Modeling Issues
  - **mean** function of GPs
  - **non-stationary covariance** of GPs

Solution: Hierarchical Gaussian Processes



- Learn the **common structure** and all the functions (what?)
- Learn a **non-stationary** GP (a parametric kernel function?)
- Learn non-stationary covariance of GP from a stationary base kernel function (**learn a GP prior in a nonparametric way!**)

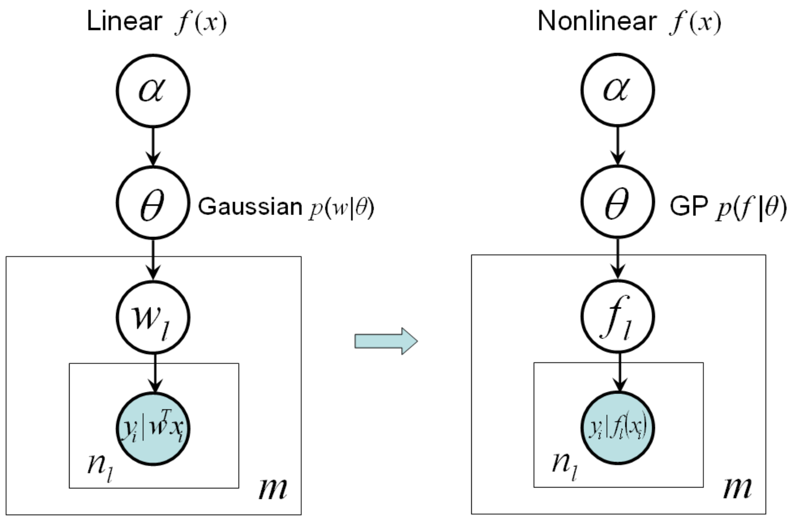
Outline

- Introduction
- Multi-task learning with linear models
- Multi-task learning with Gaussian processes
- Empirical study

Outline

- Introduction
- **Multi-task learning with linear models**
- Multi-task learning with Gaussian processes
- Empirical study

Hierarchical Linear Functions



- Simple, intuitive
- It can be generalized to nonparametric hierarchical GPs (later) kernel function (**a nonparametric way!**)

Linear Models for Multi-Task Learning

Model 1 *The multi-task linear model generates observations as follows*

1. *For task l, given $\mathbf{x}_i$, $y_i^{(l)} \sim \mathcal{N}(\mathbf{w}_l^\top \mathbf{x}_i, \sigma^2)$;*
2. *For each function $f_l = \mathbf{w}_l^\top \mathbf{x}$, $\mathbf{w}_l \sim \mathcal{N}(\boldsymbol{\mu}_w, \mathbf{C}_w)$;*
3. *$\theta = \{\boldsymbol{\mu}_w, \mathbf{C}_w\}$ follow a* ***normal-inverse Wishart (NIW) distribution***

$$\boldsymbol{\mu}_w, \mathbf{C}_w \sim \mathcal{N}(\boldsymbol{\mu}_w | \boldsymbol{\mu}_{w_0}, \frac{1}{\pi}\mathbf{C}_w)\mathcal{IW}(\mathbf{C}_w | \tau, \mathbf{C}_{w_0}). \quad (1)$$

with the hyper parameters $\pi, \tau, \mathbf{C}_{w_0} = \mathbf{I}$ and $\boldsymbol{\mu}_{w_0} = 0$.

- **Comment**: if $\pi \to \infty$ and $\tau \to \infty$, then $\mathbf{C}_w = \mathbf{I}$ and $\boldsymbol{\mu}_w = 0$, equivalent to m independent regression models;

Comments

- **Common predictive structure**: Let $\mathbf{w}_l = \boldsymbol{\mu}_w + \mathbf{v}_l$, then:
  - $\boldsymbol{\mu}_w$: the same for all the tasks
  - $\mathbf{v}_l$: different over tasks, but follow the same distribution $\mathcal{N}(0, \mathbf{C}_w)$.
- **What to learn?**
  - Estimating θ: learn the common structure over tasks.
  - Estimating $\mathbf{w}_l$: learn the functions for each tasks given the learned θ.

Maximum Penalized Likelihood Estimates

- Joint distribution: $p(\mathbf{y}_1, \ldots, \mathbf{y}_m, \mathbf{w}_1, \ldots, \mathbf{w}_m | \theta) = \prod_l \frac{1}{Z_l} \exp\big(-\frac{1}{2} J(\mathbf{w}_l)\big)$, where

$$J(\mathbf{w}_l) = \frac{1}{\sigma^2} \|\mathbf{y}_l - \mathbf{X}_l \mathbf{w}_l\|^2 + (\mathbf{w}_l - \boldsymbol{\mu}_{\mathbf{w}})^T \mathbf{C}_{\mathbf{w}}^{-1} (\mathbf{w}_l - \boldsymbol{\mu}_{\mathbf{w}})$$

- **marginalized log-likelihood**:

$$\mathcal{L}(\theta) = \ln p(\mathbf{y}_1, \ldots, \mathbf{y}_m | \theta) = \sum_l \ln \int_{\mathbf{w}_l} \frac{1}{Z_l} \exp\Big(-\frac{1}{2} J(\mathbf{w}_l)\Big) d\mathbf{w}_l$$

- **Estimates**:

$$\hat{\theta} = \arg \max_{\theta = \{\boldsymbol{\mu}_{\mathbf{w}}, \mathbf{C}_{\mathbf{w}}, \sigma\}} \mathcal{L}(\theta) + \ln p(\boldsymbol{\mu}_{\mathbf{w}}, \mathbf{C}_{\mathbf{w}})$$

Expectation-Maximization (EM)

- E-step: For each f_l, compute the sufficient statistics of $p(\mathbf{w}_l|\mathbf{D}_l, \theta)$ based on current θ.

$$\hat{\mathbf{w}}_l = \mathbf{C}_{w_l}\left(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal \mathbf{y}_l + \mathbf{C}_w^{-1}\boldsymbol{\mu}_w\right)$$
$$\mathbf{C}_{w_l} = \left(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal \mathbf{X}_l + \mathbf{C}_w^{-1}\right)^{-1}$$

- M-step: update the estimates of parameters

$$\boldsymbol{\mu}_w = \frac{1}{\pi + m}\sum_l \hat{\mathbf{w}}_l$$
$$\mathbf{C}_w = \frac{1}{\tau + m}\left\{\pi\boldsymbol{\mu}_w\boldsymbol{\mu}_w^\intercal + \tau\mathbf{I} + \sum_l \mathbf{C}_{w_l} + \sum_l \left[\hat{\mathbf{w}}_l - \boldsymbol{\mu}_w\right]\left[\hat{\mathbf{w}}_l - \boldsymbol{\mu}_w\right]^\intercal\right\}$$
$$\sigma^2 = \frac{1}{\sum_l n_l}\sum_l \|\mathbf{y}_l - \mathbf{X}_l\hat{\mathbf{w}}_l\|^2 + \mathrm{tr}[\mathbf{X}_l\mathbf{C}_{w_l}\mathbf{X}_l^\intercal]$$

Outline

- Introduction
- Multi-task learning with linear models
- **Multi-task learning with Gaussian processes**
- Empirical study

From Linear Models to GPs

- If $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathrm{w}}, \mathbf{C}_{\mathrm{w}})$, then a GP is defined with
  - mean function $\mu = \mathrm{E}[f(\mathbf{x})] = \boldsymbol{\mu}_{\mathrm{w}}^T \mathbf{x}$
  - covariance function $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{C}_{\mathrm{w}} \mathbf{x}'$
- **Implicit feature mapping**: let $\mathbf{C}_{\mathrm{w}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, it is easy to see $K(\mathbf{x}, \mathbf{x}') = \langle \Phi(\mathbf{x}), \Phi(\mathbf{x}') \rangle$, where $\big(\Phi(\mathbf{x})\big)_k = \sqrt{\lambda_k} \langle \mathbf{x}, \mathbf{u}_k \rangle$;
- The connection suggests that we can solve the problem in a **nonparametric** way, namely by directly estimating the mean and kernel of a function space.

A Wishart Prior for GPs

- For $\mathbf{f}_l = [f_l(\mathbf{x}_1), \ldots, f_l(\mathbf{x}_n)]^\top$ realized on **any finite set** $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$, it can be proven that our linear model equivalently specifies
  - $\mathbf{f}_l \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{f}, \mathbf{K})$
  - $\boldsymbol{\mu}_\mathrm{f}, \mathrm{K}$ **also follow an NIW distribution** $\mathcal{N}(\boldsymbol{\mu}_\mathbf{f}|0, \frac{1}{\pi}\mathbf{K})\mathcal{IW}(\mathbf{K}|\tau, \boldsymbol{\kappa})$

where $\boldsymbol{\mu}_\mathbf{f} = \boldsymbol{\mu}_\mathbf{w}^\top \mathbf{X}$, $\mathbf{K} = \mathbf{X}\mathbf{C}_w\mathbf{X}^\top$ and $\boldsymbol{\kappa}_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$

Transductive Multi-Task GPs

Model 2 *(Transductive Model) Let* $\mathbf{f}^l$ *be the values of* f_l *on a set* $\mathbf{X}$*, the generative model is as the following*

1. $\boldsymbol{\mu}_\mathbf{f}, \mathbf{K} \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{f}|0, \frac{1}{\pi}\mathbf{K})\mathcal{IW}(\mathbf{K}|\tau, \boldsymbol{\kappa})$;
2. *For each function* f_l, $\mathbf{f}^l \sim \mathcal{N}(\boldsymbol{\mu}_\mathbf{f}, \mathbf{K})$;
3. *Given* $\mathbf{x}_i \in \mathbf{X}_l$, $y_i^l = \mathbf{f}_i^l + \epsilon$ *where* $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

- It can be again solved by **EM algorithm**.
- **Nonlinear functions** obtained if a nonlinear base kernel function $\kappa(\cdot, \cdot)$ is chosen;

Comments

- The model is equivalent to our linear model, but focuses on finite number of data points;
- **Kernel Learning**: A kernel matrix $\mathbf{K}$ is adapted from a base kernel function $\kappa(\cdot,\cdot)$;
- $\mathbf{K}$ can be expanded to include any new test points, as long as the base kernel $\kappa(\cdot,\cdot)$ on them has been evaluated;
- **How to make predictions on new test points without retraining?**

Duality of NIW Distribution

- Given $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\top$ and $\boldsymbol{\kappa} \succ 0$, there exists a unique $\boldsymbol{\alpha} \in \mathbb{R}^n$ such that, $\mathbf{f} = \boldsymbol{\kappa}\boldsymbol{\alpha}$
- Then we can prove
  - $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha)$
  - $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ **follow a NIW distribution with scale matrix $\boldsymbol{\kappa}^{-1}$**:

$$p(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha) = \mathcal{N}(\boldsymbol{\mu}_\alpha | 0, \frac{1}{\pi}\mathbf{C}_\alpha)\mathcal{IW}(\mathbf{C}_\alpha | \tau, \boldsymbol{\kappa}^{-1}) \tag{2}$$

Comments: we can equivalently work on a generative model of weights $\boldsymbol{\alpha}_l$ for $\mathbf{f}_l = \boldsymbol{\kappa}\boldsymbol{\alpha}_l$.

Inductive Multi-Task Learning

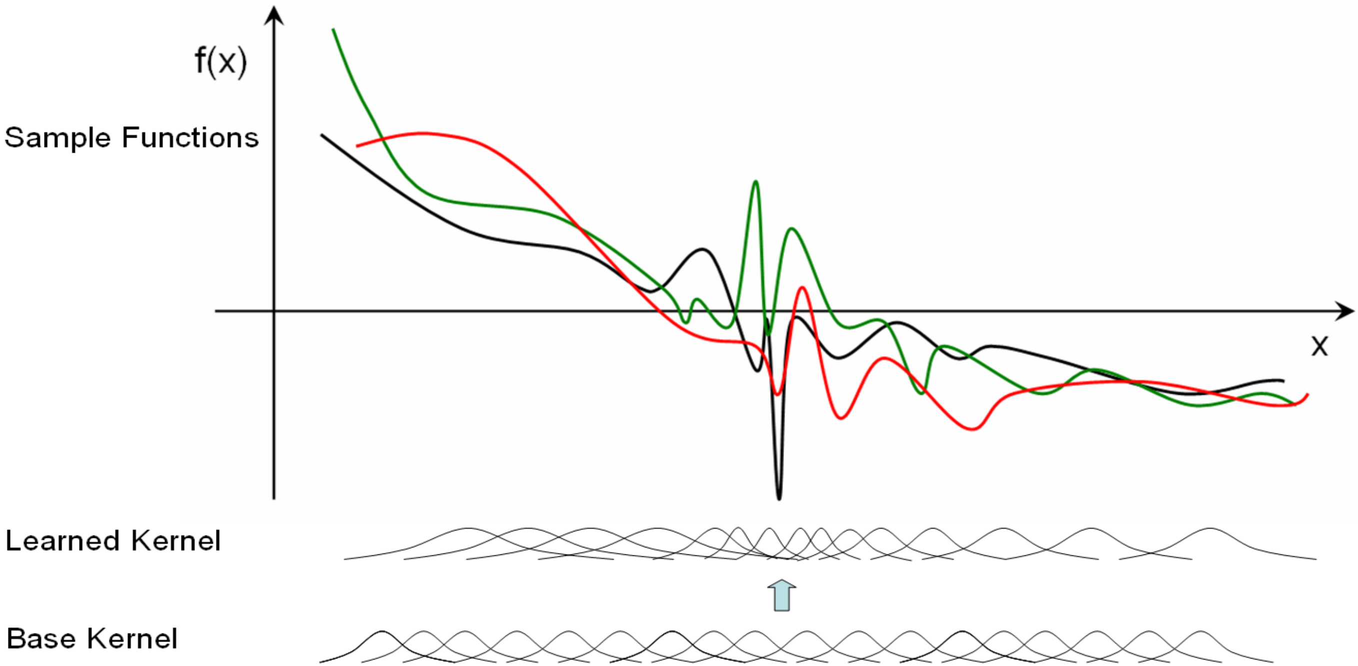
Model 3 *(Inductive Model) Let* $\mathbf{f}^l$ *be the values of* f_l *on a set* $\mathbf{X}$*, satisfying* $\cup \mathbf{X}_l \subseteq \mathbf{X}$*. the generative model is defined as:*

1. $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ *are generated once* (2);
2. *For each function* f_l, $\boldsymbol{\alpha}^l \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha)$;
3. *Given* $\mathbf{x} \in \mathbf{X}_l$, $y = \sum_{i=1}^{n} \alpha_i^l \kappa(\mathbf{x}_i, \mathbf{x}) + \epsilon$ *where* $\epsilon \sim \mathcal{N}(0, \sigma^2)$, $\mathbf{x}_i \in \mathbf{X}$.

Finite Dimensionality of Mean Predictions

- **Good News**:
  - **Theorem**: we get exactly the correct predicted mean function (independent to unlabeled points in $\boldsymbol{\kappa}$)
- **Bad News**:
  - The predictive variance for new test point cannot be fully explored (just a Schur complement)
- To have full predictive variance on a new point, we have to incorporate this point into $\boldsymbol{\kappa}$, and retrain the model (efficient way to do this?)

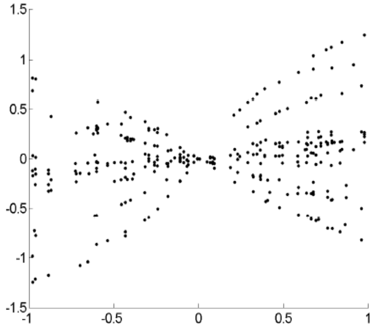
Summarization of the Idea



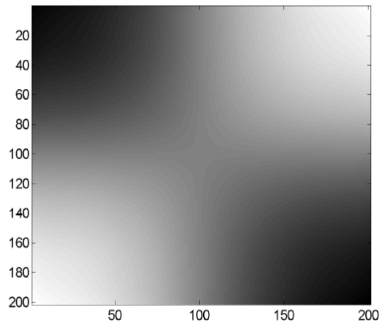
Outline

- Introduction
- Multi-task learning with linear models
- Multi-task learning with Gaussian processes
- **Empirical study**

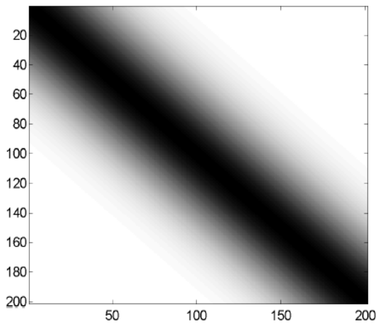
A Toy Problem



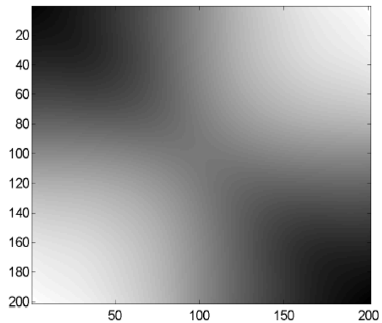
(a) Toy Data



(b) True Kernel Matrix



(c) Base Kernel Matrix



(d) Learned Kernel Matrix

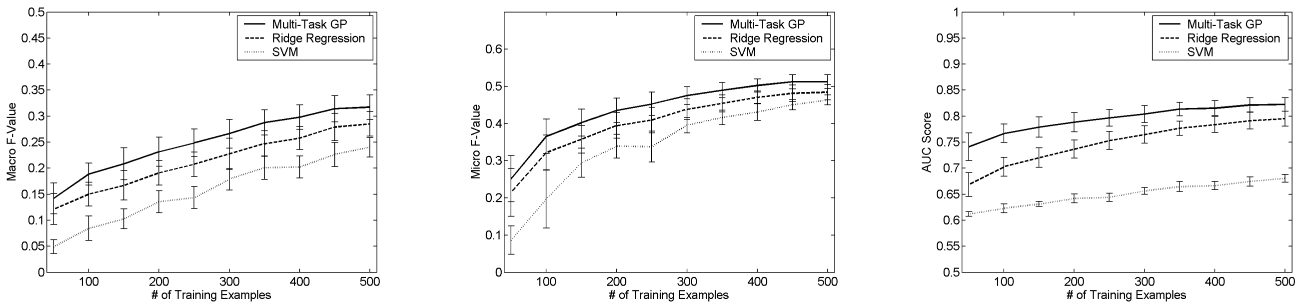
Multi-Label Text Categorization (I)

Table 1: Text Categorization on RCV1

	ALL			PARTIALLY LABELED		
	AUC	F-MICRO	F-MACRO	AUC	F-MICRO	F-MACRO
MULTI-TASK GP	0.773	0.605	0.260	0.826	0.623	0.281
RIDGE REGRESSION	0.756	0.584	0.245	0.771	0.564	0.240
SVM	0.697	0.573	0.221	0.716	0.547	0.212

- Training set: fixed 50 categories, 10 random repeats to choose 1000 documents, 300 random labeled examples for each category
- Test set: 10000 documents

Multi-Label Text Categorization (II)



(*a*) (*b*) (*c*)

Generalization of learned kernels on other 31 categories (each measure averaged over 50 repeats)

Conclusions

- Suggest a **novel Bayesian multi-task framework** to overcome the drawbacks of reported methods
  - capture both the first and second order dependency of functions;
  - can handle nonlinear functions
  - generalizable to new test points
- Explore the equivalence between parametric linear approaches and **nonparametric GP approaches** to multi-task learning
  - The duality of Wishart distribution
- Suggest a **new kernel learning** framework based on a base kernel function

Related Work: Bayesian Methods

Parametric ...

- Bayesian multi-task learning [Bakker & Heskes, 2003]: parametric, easily overfitting since no control (prior) for θ.
- Conjoint Analysis [Chapelle & Harchaoui 2005]: Similar to our model in the linear case, not capable to handle nonlinear functions.

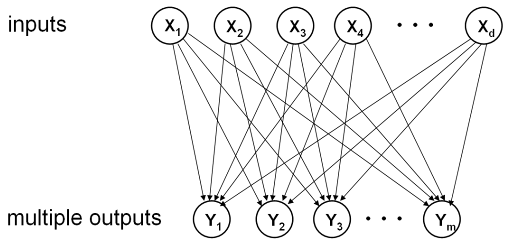
Nonparametric ...

- Learning to learn with IVM [Lawrence & Platt 2004]: Explore the sparsity of the common predictive structure, to reduce the computational complexity.
- Learn GPs via Hierarchical Bayes [Schwaighofer, Tresp & Yu, 2005] Learning multiple functions defined on fixed inputs, needs additional step to handle new test points.

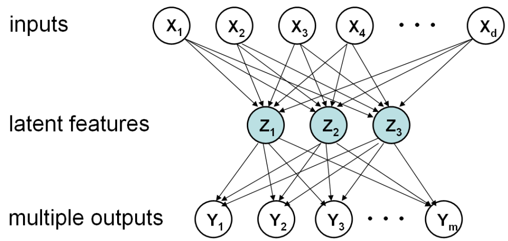
Related Work: Non-Bayesian Methods

- Regularized multi-task Learning [Evgeniou & Pontil 2004]:
  - Learning multiple linear functions: $f_l(\mathbf{x}) = \mathbf{w}_l^T\mathbf{x}$, $l = 1, \ldots, m$;
  - Let $\mathbf{w}_l = \mathbf{w}_0 + \mathbf{v}_l$, where $\mathbf{w}_0$ is unchanged over functions, while $\mathbf{v}_l$ are independent of each other;
  - Only consider the **mean effect** of functions
- Learning predictive structure from multiple tasks [Ando & Zhang, 2005]: iterative alternating optimization, at each step, first estimate $\mathbf{w}_1, \ldots, \mathbf{w}_m$, and then perform PCA on $\mathbf{W} = [\mathbf{w}_1, \ldots, \mathbf{w}_m]$, use the leading k eigenvectors to constrain new estimates of $\mathbf{w}_1, \ldots, \mathbf{w}_m$;
  - Seems to model **covariance**, but dimensionality has to be chosen
  - Unclear how to handle nonlinear functions

A Parametric View



independent tasks



dependent tasks

- The latent space captures the **common structure**.
- One way to do supervised feature learning.
- Any **nonparametric treatment?**

A Function Space View

- For each task

$$\min_{f_l \in \mathcal{H}_\theta} \sum_i \ell\big(f_l(\mathbf{x}_i), y_i\big) + \lambda \|f_l\|^2_{\mathcal{H}_\theta}$$

- Optimize the **function space** $\mathcal{H}_\theta$, which captures the common structure of tasks
- Unclear what objective function to optimize

Thanks! Questions? Suggestions?