Learning GPs from Multiple Tasks

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

SIEMENS [1] Corporate Technology, Siemens, Munich

FIRST [2] Intelligent Data Analysis, Fraunhofer FIRST, Berlin

Multi-Task Learning

- Learn a set of different but **related** predictive problems.
- Instead of separated training, **solve them jointly!**
- Exploring the **statistical dependency** between tasks.

Multi-Task Learning

- Learn a set of different but **related** predictive problems.
- Instead of separated training, **solve them jointly!**
- Exploring the **statistical dependency** between tasks.
- **How trivial it is applying GPs to solve the problem!**

Examples to Motivate ...

- **Multi-label text categorization**: One document can belong to more than one categories — categories are semantically related.
- **Collaborative filtering**: Predicting many users preference jointly, instead of treating them separately — people's opinions are influenced by each other.
- **Computer vision**: Tracking the movement of different parts of a robot — mutually constrained freedoms.

Learn the Predictive Function

- **Single-Task Learning**: From a function space $\mathcal{H}$, to pick the function $f \in \mathcal{H}$ that has low complexity $\|f\|_{\mathcal{H}}^2$ and meanwhile explains empirical data very well

$$\min_{f \in \mathcal{H}} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{D}} \ell\big(f(\mathbf{x}_i), y_i\big) + \lambda \|f\|_{\mathcal{H}}^2$$

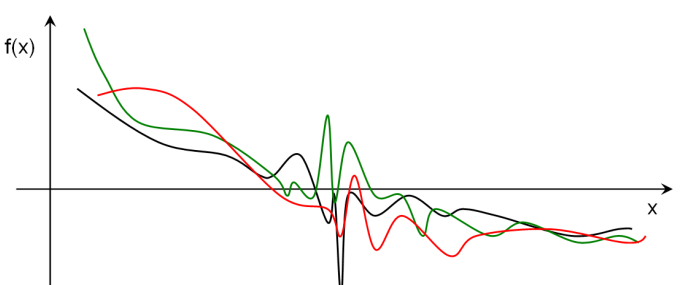
where $\ell(\cdot, \cdot)$ is an empirical loss.

Learn the Common Structure

- **Multi-Task Learning**: learn many functions together, and also optimize the function space $\mathcal{H}_\theta$ to make it suitable for all the functions

$$\min_{\theta} \left\{ \sum_{l} \min_{f_l \in \mathcal{H}_\theta} \sum_{(\mathbf{x}_i, y_i) \in \mathbf{D}_l} \ell\big(f_l(\mathbf{x}_i), y_i\big) + \lambda \| f \|^2_{\mathcal{H}_\theta} \right\} + \gamma \eta(\theta)$$

where θ captures common structure shared by all the functions.

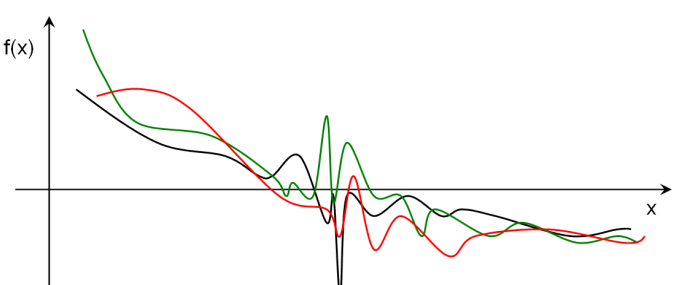


Bayesian Approaches

- **Hierarchical Bayes**: all the functions are generated from a common prior distribution

$$p(\{\mathbf{y}_l\}|\{\mathbf{X}_l\},\theta) = \prod_l \int p(\mathbf{y}_l|f_l,\mathbf{X}_l)p(f_l|\theta)df_l$$

where $p(f|\theta)$ captures common structure shared by all the functions.



Related Work

- Bayesian multi-task learning [Bakker and Heskes, 2003]: parametric, easily overfitting since no control for θ.
- Learning to learn with IVM [Lawrence and Platt 2004]: Explore the sparsity of the common predictive structure, to reduce the computational complexity.
- Regularized multi-task Learning [Evgeniou and 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 $\mathbf{w}_0$ models the **mean effects** of functions, while $\mathbf{v}_l$ are independent of each other;
- Learning predictive structure from multiple tasks [Ando and Zhang, 2005]: an iterative algorithm, 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 capture the common **covariance** structure;

Our Approaches

- In contrast to the two recent frequentist approaches that applied linear models, Bayesian treatments appear to be a more natural and more general framework;
- Derive a nonparametric framework by exploring the duality of linear models and GPs;
- Propose a general kernel learning framework, based on the infinite-dimensional Wishart distribution.

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

Settings for Multi-Task Learning

- Consider m predictive learning tasks indexed as $l = 1, \ldots, m$.
- For each task l we have observed n_l labeled examples $\mathbf{D}_l = (\mathbf{X}_l, \mathbf{y}_l)$, where $\mathbf{X}_l \in \mathbb{R}^{n_l \times d}$ and $\mathbf{y}_l \in \mathbb{R}^{n_l}$;
- The goal is to learn m functions $f_l(\mathbf{x}) = \mathbf{w}_l^T \mathbf{x}$ that explain the data.

The Basic Idea

- Instead of fixing $p(\mathbf{w}) = \mathcal{N}(0, \mathbf{I})$ *a priori*, we try to learn $p(\mathbf{w}) = N(\boldsymbol{\mu}_w, \mathbf{C}_w)$;
- To obtain robust estimation for parameters $\theta = \{\boldsymbol{\mu}_w, \mathbf{C}_w\}$, we adopt a **normal-inverse Wishart (NIW) distribution** to constrain the freedom of them

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

Linear Models for Multi-Task Learning

Model 1 *Given* $p(\boldsymbol{\mu}_w, \mathbf{C}_w)$ *with the hyper parameters* $\pi, \tau, \mathbf{C}_{w_0} = \mathbf{I}$ *and* $\boldsymbol{\mu}_{w_0} = 0$*, define the generative model:*

1. $\boldsymbol{\mu}_w, \mathbf{C}_w$ *are sampled once from* $p(\boldsymbol{\mu}_w, \mathbf{C}_w)$ *given by* (1)*;*
2. *For each function* f_l, $\mathbf{w}_l \sim \mathcal{N}(\boldsymbol{\mu}_w, \mathbf{C}_w)$*;*
3. *Given* $\mathbf{x}_i \in \mathbf{X}_l$, $y_i^l = \mathbf{w}_l^\intercal \mathbf{x} + \epsilon$ *where* $\epsilon \sim \mathcal{N}(0, \sigma^2)$.

Comments:

- when $\pi \to \infty$ and $\tau \to \infty$, the model becomes identical to m independent regression models since $\mathbf{C}_w = \mathbf{I}$ and $\boldsymbol{\mu}_w = 0$;
- With intermediate τ and π, we can control how much θ is adapted to the empirical data;
- $\mathbf{C}_w$ reflects an implicit mapping of original features $\mathbf{x}$.

What's the Difference?

- **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 constrained by the same covariance.
- **Two-stage** learning procedure:
  - Estimating θ: learn the common structure over tasks.
  - Estimating $\mathbf{w}_l$: learn the functions for each tasks given the learned θ.

Joint Distribution

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

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

Maximum Penalized Likelihood Estimates

- **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.

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

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

Expectation-Maximization (EM)

- M-step: update the estimates of parameters

$$\boldsymbol{\mu}_w = \frac{1}{\pi + m} \sum_l \hat{\mathbf{w}}_l$$

$$\mathbf{C}_w = \frac{1}{\tau + m} \left\{ \pi \boldsymbol{\mu}_w \boldsymbol{\mu}_w^\intercal + \tau \mathbf{I} + \sum_l \mathbf{C}_{w_l} + \sum_l \left[\hat{\mathbf{w}}_l - \boldsymbol{\mu}_w \right] \left[\hat{\mathbf{w}}_l - \boldsymbol{\mu}_w \right]^\intercal \right\}$$

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

Outline

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

From Linear Models to GPs

- If $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_{\mathbf{w}}, \mathbf{C}_{\mathbf{w}})$, then a GP is defined with
  - mean function $\mu = \mathrm{E}[f(\mathbf{x})] = \boldsymbol{\mu}_{\mathbf{w}}^T \mathbf{x}$
  - covariance function $K(\mathbf{x}, \mathbf{x}') = \mathbf{x}^T \mathbf{C}_{\mathbf{w}} \mathbf{x}'$
- **Implicit feature mapping**: let $\mathbf{C}_{\mathbf{w}} = \mathbf{U}\boldsymbol{\Lambda}\mathbf{U}^T$, 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 directly estimate the mean and kernel of a function space.

The Hyperprior for $p(f)$

Theorem 1 *Let $\mathcal{S} \subset \mathbb{R}^d$ be a set of data points, $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{S}$, $\kappa(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$ defines a positive definite kernel. Then for **any** given subset of points $\mathbf{X} = [\mathbf{x}_1, \ldots, \mathbf{x}_n]$ in $\mathcal{S}$, Model.1 equivalently specifies a prior distribution for the mean $\boldsymbol{\mu}_\mathbf{f}$ and the covariance $\mathbf{K}$ of function values $\mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\intercal$, which is a normal-inverse-Wishart distribution,*

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

where $\boldsymbol{\kappa} \succ 0$ with $\boldsymbol{\kappa}_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$

Comments: In particular, if $\mathcal{S}$ is an infinite-dimensional space, for f realized on **any finite set of inputs**, its mean and covariance follow an NIW with the corresponding base matrix $\boldsymbol{\kappa}$.

An infinite-dim NIW, or called NIW processes?

Transductive Multi-Task Learning

Model 2 *(Transductive Model) Let* $\mathbf{f}^l$ *be the values of* f_l *on a set* $\mathbf{X}$*, satisfying* $\cup \mathbf{X}_l \subseteq \mathbf{X}$*. Given the hyper prior distribution described in* (2)*, define as the generative model:*

1. $\boldsymbol{\mu}_{\mathbf{f}}, \mathbf{K}$ *are sampled once from the hyper prior;*
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.

Connections to Model 1

- The model is equivalent to Model 1, but focuses on finite number of data points. For functions $\{f_l\}$ defined on a finite set $\mathbf{X}$, we can learn the corresponding mean of functions and the kernel matrix $\mathbf{X}$;
- $\mathbf{X}$ can be expanded by including arbitrary test points, as long as the base kernel $\kappa(\cdot, \cdot)$ on them can be evaluated;
- With nonlinear base kernel $\kappa(\cdot, \cdot)$, we can now handel nonlinear functions;
- How to do **inductive** multi-task learning?

Duality of NIW Distribution

Theorem 2 *Given* $\boldsymbol{\mu}_{\mathbf{f}}$ *and* $\mathbf{K}$ *sampled from the hyper prior specified in* (2)*, there exist unique* $\boldsymbol{\mu}_\alpha \in \mathbb{R}^n$ *and* $\mathbf{C}_\alpha \in \mathbb{R}^{n\times n}$ *such that*

1. $\boldsymbol{\mu}_{\mathbf{f}} = \boldsymbol{\kappa}\boldsymbol{\mu}_\alpha, \quad \mathbf{K} = \boldsymbol{\kappa}\mathbf{C}_\alpha\boldsymbol{\kappa}$
2. $\forall \mathbf{f} = [f(\mathbf{x}_1), \ldots, f(\mathbf{x}_n)]^\intercal$*, there exists a unique* $\boldsymbol{\alpha} \in \mathbb{R}^n$ *such that,* $\mathbf{f} = \boldsymbol{\kappa}\boldsymbol{\alpha}$ *and* $\boldsymbol{\alpha} \sim \mathcal{N}(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha)$
3. $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ *follow a NIW distribution with scale matrix* $\boldsymbol{\kappa}^{-1}$*:*

$$p(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha) = \mathcal{N}(\boldsymbol{\mu}_\alpha|0, \frac{1}{\pi}\mathbf{C}_\alpha)\mathcal{IW}(\mathbf{C}_\alpha|\tau, \boldsymbol{\kappa}^{-1}) \tag{3}$$

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* $\cup \mathbf{X}_l \subseteq \mathbf{X}$*. Given the hyper prior distribution of* $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ *described in theorem 2, define as the generative model:*

1. $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ *are generated once (3);*
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* $\epsilon \sim \mathcal{N}(0, \sigma^2)$*,* $\mathbf{x}_i \in \mathbf{X}$*.*

In Which Sense Inductive?

Theorem 3 *Suppose a finite set* $\mathbf{X}$ *is given, satisfying* $\cup \mathbf{X}_l \subseteq \mathbf{X}$. *Let* $S \subset \mathbb{R}^d$ *be the subspace spanned by the columns of* $\mathbf{X}$ *and* $\mathbf{P}$ *be the orthogonal projection onto* S. *If there is a constraint* $\mathbf{w} = \mathbf{P}\mathbf{w}'$ *and* $\mathbf{w}' \sim \mathcal{N}(\boldsymbol{\mu}_w, \mathbf{C}_w)$ *in Model 1, then the following conclusions hold:*

1. *The modified Model 1 is equivalent to Model 3;*
2. *The estimates* $\hat{\mathbf{w}}_l$, $l = 1, \ldots, m$, *in Model 1 are invariant to the modification.*

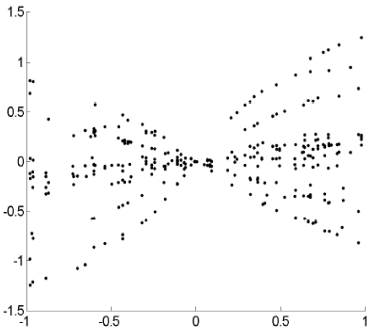
Comments

- Model 3 and Model 1 produce exactly the same estimates of $\{f_l\}$;
- Model 3 and Model 1 produce different predictive variances on new test points $x \notin \mathbf{X}$.
- Somewhat like representor theorem.

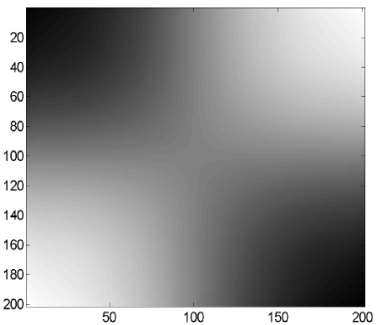
Outline

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

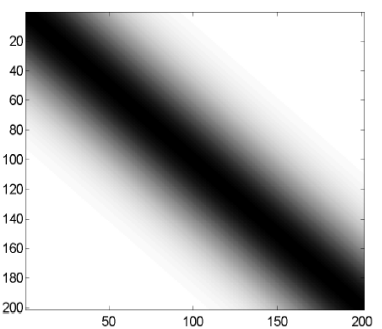
A Toy Problem



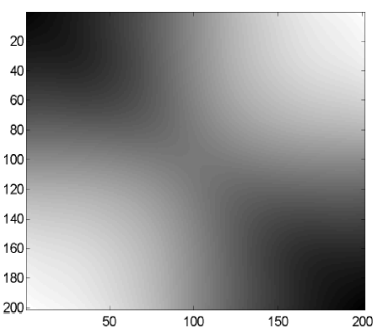
(a) Toy Data



(b) True Kernel Matrix

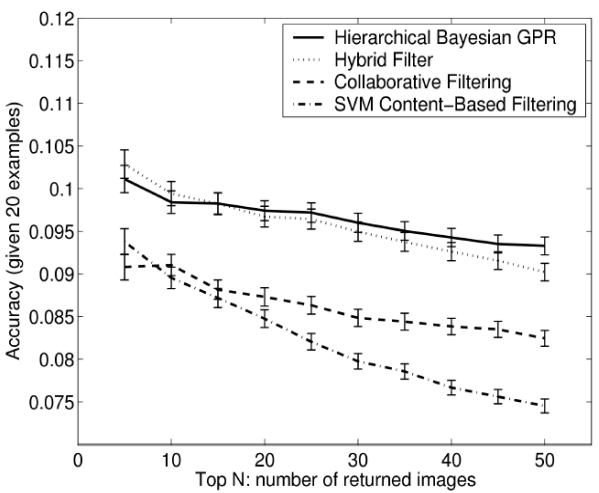


(c) Base Kernel Matrix

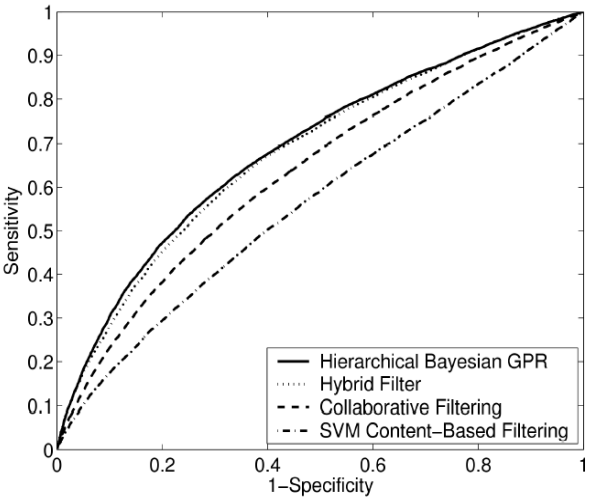


(d) Learned Kernel Matrix

Predict User Preferences



Top-20 accuracy



ROC curves

190 users' preferences (like or dislike) on 640 paintings. For each user we pick up 20 examples for training and predict the rests.

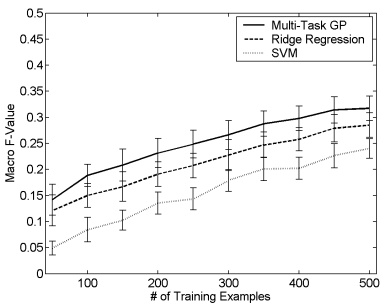
Multi-Label Text Categorization (I)

Table 1: Text Categorization on RCV1

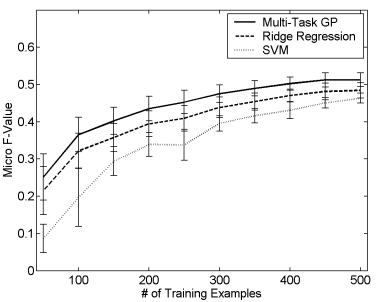
	ALL			Partially Labeled		
	AUC	F-micro	F-macro	AUC	F-micro	F-macro
Multi-Task GP	0.773	0.605	0.260	0.826	0.623	0.281
Ridge Regression	0.756	0.584	0.245	0.771	0.564	0.240
SVM	0.697	0.573	0.221	0.716	0.547	0.212

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

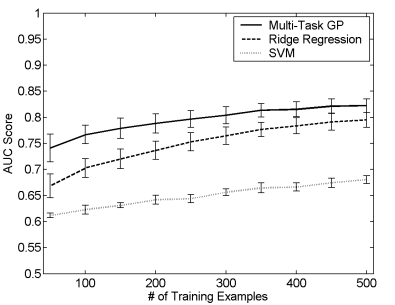
Multi-Label Text Categorization (II)



(*a*)



(*b*)



(*c*)

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

Summarization

- A linear model for multi-task learning was naturally derived from conventional regularized linear models. Compared to related work, our approach is more general in the sense both mean and covariance of function weights are explored.
- A nonparametric framework for multi-task learning was built upon the linear models in high or infinite dimensional space. The connections underly a direct definition of hyper prior $p(\boldsymbol{\mu}, \mathbf{K})$ for the hypothesis space $p(f|\boldsymbol{\mu}, \mathbf{K})$.
- A Bayesian treatment for learning a kernel based on base kernel functions, $\kappa(\cdot, \cdot)$, which defines the feature space, while the learned kernel $K(\cdot, \cdot)$ reflects implicit linear mapping of features ....
- This morning Tony O'Hagan's talk, latent variable model for non-stationary spatial modeling.

Thanks! Questions? Suggestions?