Semi-Supervised Multi-Task Regression

Yu Zhang and Dit-Yan Yeung

Hong Kong University of Science and Technology
{zhangyu,dyyeung}@cse.ust.hk

Abstract. Labeled data are needed for many machine learning applications but the amount available in some applications is scarce. Semi-supervised learning and multi-task learning are two of the approaches that have been proposed to alleviate this problem. In this paper, we seek to integrate these two approaches for regression applications. We first propose a new supervised multi-task regression method called SMTR, which is based on Gaussian processes (GP) with the assumption that the kernel parameters for all tasks share a common prior. We then incorporate unlabeled data into SMTR by changing the kernel function of the GP prior to a data-dependent kernel function, resulting in a semi-supervised extension of SMTR, called SSMTR. Moreover, we incorporate pairwise information into SSMTR to further boost the learning performance for applications in which such information is available. Experiments conducted on two commonly used data sets for multi-task regression demonstrate the effectiveness of our methods.

1 Introduction

Many machine learning applications require that labeled data be available for model training. Unfortunately the amount of labeled data available in some applications is very scarce because labeling the data points manually is very tedious and costly. As a consequence, the model thus learned is often not satisfactory in performance. To alleviate this problem, machine learning researchers have investigated various approaches, with *semi-supervised learning* and *multi-task learning* being two popular ones.

Semi-supervised learning [1] can be seen as an extension of the conventional supervised learning paradigm by augmenting the (labeled) training data set with unlabeled data so as to exploit the useful information in the unlabeled data to boost learning performance. Early semi-supervised learning methods include co-training [2], which builds two learning models based on two different views of the data and then uses each learning model to select confident unlabeled data for the other, and transductive SVM [3, 4], which uses both labeled and unlabeled data to maximize the margin of a support vector machine (SVM). More recent development includes many graph-based methods [5–7], which model the geometric relationship between all data points in the form of a graph and then propagate the label information from the labeled data points to the unlabeled data points throughout the graph. In order for the unlabeled data to be useful for semi-supervised learning, some assumptions about the data have to be satisfied. Two widely used assumptions are the cluster assumption and manifold assumption. The cluster assumption simply means that if two points are in the same cluster, they are more likely to belong to the same class. Equivalently, this means that the class decision boundary only

goes through low-density regions. Transductive SVM is one popular method based on the cluster assumption. As for the manifold assumption, it means that the data points in some high-dimensional space span a low-dimensional manifold. If two points are close to each other with respect to some metric on the manifold, their outputs are likely to be similar. As a result, if we want to preserve the manifold structure when performing some projection, then two points that are close in the manifold should remain close after projection. The manifold assumption is the underlying model assumption of many semi-supervised learning methods, particularly graph-based methods.

On the other hand, multi-task learning [8–10] seeks to improve the learning performance of one task with the help of some related tasks. This approach has been inspired by psychological observations that humans can often benefit from previous learning experience when learning a new but related task, sometimes referred to as *transfer of learning*. Many multi-task learning methods have been proposed over the past decade. For example, multi-task feature learning [11] learns a common representation for all tasks under the regularization framework, regularized multi-task SVM [12] extends SVM by requiring that the SVM parameters for all tasks be close to each other, task clustering methods [13, 14] group the tasks into multiple clusters and then learn a similar or common representation for all tasks within a cluster, and GP-based multi-task learning methods [15–18] utilize Gaussian processes (GP) as the base model for multi-task learning. For multi-task learning, many existing methods assume that all the tasks are related to each other and hence similar or identical data features or model parameters are shared by all tasks or subsets of tasks, for example, all tasks in the same cluster. Methods based on neural networks and multi-task feature learning all assume that the data features are shared by all tasks. On the other hand, regularized multi-task SVM and the methods in [14] assume that similar model parameters are shared by all tasks or tasks in the same cluster. Moreover, some methods incorporate both assumptions in their models, e.g., [13].

Since semi-supervised learning and multi-task learning share the common objective of seeking to improve the learning performance of the original supervised learning task by exploiting some auxiliary data available (unlabeled data for the current task or labeled data for other related tasks), it makes sense to combine them in an attempt to get the best of both worlds. Indeed, some such attempts have been made recently. The method proposed by Ando and Zhang [19] bears some relationship with these two learning paradigms even though its objective is mainly to improve the performance of semi-supervised learning. There exists only a single task (target task) to start with as well as some unlabeled data. The unlabeled data are then utilized to create more tasks to help the learning of the target task. Moreover, Liu et al. [20] proposed a semi-supervised learning method called parameterized neighborhood-based classification (PNBC), which applies random walk to logistic regression and uses a task clustering method for multi-task learning. However, these two methods only consider the classification problem and cannot be extended readily to the regression problem. Indeed, there exist some applications that can be modeled as the combination of semi-supervised regression and multi-task regression, for example, personalized pose estimation. In personalized pose estimation, each task corresponds to the pose estimation

for one person. In this application, there exist large amount of images with unknown pose information for each person.

To the best of our knowledge, there does not exist any work in the literature that integrates semi-supervised regression and multi-task regression. In this paper, we want to fill the gap by proposing a scheme for such integration. We first propose a new supervised multi-task regression method called SMTR, which is based on GP with the assumption that the kernel parameters for all tasks share a common Gaussian prior. We then incorporate unlabeled data into SMTR by changing the kernel function of the GP prior to a data-dependent kernel function, resulting in a semi-supervised extension of SMTR, called SSMTR. Moreover, as in [21], we incorporate pairwise information into SSMTR to further boost the learning performance for applications in which such information is available.

We first present SMTR in Section 2. SSMTR, our semi-supervised extension of SMTR, and the incorporation of pairwise information into SSMTR are then presented in Sections 3 and 4, respectively. Section 5 reports some experimental results to provide empirical evaluation of our proposed methods.

2 Supervised Multi-Task Regression

Let there be m related regression tasks $T_1, \ldots, T_m$. For task T_i, the training set D_i consists of n_i labeled data points $\{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{n_i}$ with the jth point $\mathbf{x}_j^i \in \mathbb{R}^d$ and its output $y_j^i \in \mathbb{R}$.

For each task, we use a GP [22] as the base regressor. For task T_i, we define a latent variable f_j^i for each data point $\mathbf{x}_j^i$. The prior of $\mathbf{f}^i$ is defined as

$$\mathbf{f}^i \,|\, \mathbf{X}^i \sim \mathcal{N}(\mathbf{0}_{n_i}, \mathbf{K}_{\boldsymbol{\theta}_i}), \tag{1}$$

where $\mathbf{f}^i = (f_1^i, \ldots, f_{n_i}^i)^T$, $\mathbf{X}^i = (\mathbf{x}_1^i, \ldots, \mathbf{x}_{n_i}^i)$, $\mathcal{N}(\mathbf{m}, \boldsymbol{\Sigma})$ denotes a multivariate Gaussian distribution with mean $\mathbf{m}$ and covariance matrix $\boldsymbol{\Sigma}$, $\mathbf{0}_{n_i}$ denotes an $n_i \times 1$ zero vector, and $\mathbf{K}_{\boldsymbol{\theta}_i}$ denotes the kernel matrix defined on $\mathbf{X}^i$ where the kernel function is parameterized by $\boldsymbol{\theta}_i$.

The likelihood for each task T_i is defined based on the Gaussian noise model:

$$\mathbf{y}^i \,|\, \mathbf{f}^i \sim \mathcal{N}(\mathbf{f}^i, \sigma^2 \mathbf{I}_{n_i}), \tag{2}$$

where $\mathbf{y}^i = (y_1^i, \ldots, y_{n_i}^i)^T$, σ^2 denotes the noise level, and $\mathbf{I}_{n_i}$ is the $n_i \times n_i$ identity matrix.

Since all tasks are assumed to be related, we impose a common prior on the kernel parameters $\{\boldsymbol{\theta}_i\}_{i=1}^m$ for all m tasks:

$$\boldsymbol{\theta}_i \sim \mathcal{N}(\mathbf{m}_\theta, \boldsymbol{\Sigma}_\theta). \tag{3}$$

The graphical model for SMTR is depicted in Figure 1.

In some formulation of GP regression, the noise level σ^2 can also be regarded as one element of the kernel parameters $\boldsymbol{\theta}_i$ since GP regression has an analytical form for $p(y_j^i \,|\, \mathbf{x}_j^i)$. So the noise levels for different tasks can also share a common prior as

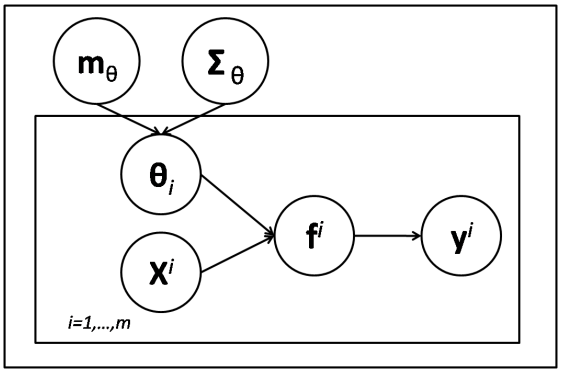

Fig. 1. Graphical model for Supervised Multi-Task Regression.

in Eq. (3) but they are not identical. Note that the noise level can be estimated from the labeled data. Since the number of labeled data points in semi-supervised learning is typically not very large, it may not be possible to obtain an accurate estimate of the noise level if estimation is done independently for each task based on limited labeled data. For this reason, we assume in this paper that all tasks have the same noise level. The more general case that allows different noise levels for different tasks will be studied in the future.

There exist some GP-based multi-task regression models [15–18]. Lawrence and Platt [15] proposed a multi-task regression model in which the kernel parameters are shared by all tasks. This assumption becomes unreasonable when there exist outlier tasks. This problem also exists in the models of [16] and [17], which later motivated the development of a robust model using t-processes [23]. Unlike the model in [15], the models in [16, 17] learn the kernel matrix in a nonparametric way. This makes it difficult to perform inductive inference since there is no parametric form for the kernel function. Bonilla et al. [18] proposed a powerful multi-task GP regressor which is especially suitable for multi-output regression problems. Their method directly models the similarity between multiple tasks and is equivalent to using a matrix-variate normal distribution to model the multiple latent function values. Due to the use of the Kronecker product, this method incurs high storage and computational costs. However, these difficulties do not exist in our proposed model. In our model, the kernel parameters for different tasks just share the same prior but are not identical, making it capable of modeling outlier tasks. Our model has a parametric form for the kernel function and hence it can be used to make inductive inference directly. Even though our model does not directly characterize the relatedness between tasks, it is implicitly characterized by the kernel parameters. Moreover, since the dimensionality of $\boldsymbol{\theta}_i$ is usually not very large, the storage cost is not high. Although there exist some multi-task learning methods which also place a common prior on the model parameters of different tasks [24, 25, 13], to the best of our knowledge none of them is based on GP.

2.1 Learning and Inference

Since

$$\begin{aligned} p(\mathbf{y}^i|\mathbf{X}^i) &= \int p(\mathbf{y}^i|\mathbf{f}^i)p(\mathbf{f}^i|\mathbf{X}^i)d\mathbf{f}^i \\ &= \mathcal{N}(\mathbf{y}^i|\mathbf{0}_{n_i}, \mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2\mathbf{I}_{n_i}), \end{aligned}$$

the log-likelihood of all tasks can be computed as

$$\begin{aligned} L = &-\frac{1}{2}\sum_{i=1}^{m}\left[(\mathbf{y}^i)^T(\mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2\mathbf{I}_{n_i})^{-1}\mathbf{y}^i + \ln\left|\mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2\mathbf{I}_{n_i}\right|\right] \\ &-\frac{1}{2}\sum_{i=1}^{m}\left[(\boldsymbol{\theta}_i - \mathbf{m}_\theta)^T\boldsymbol{\Sigma}_\theta^{-1}(\boldsymbol{\theta}_i - \mathbf{m}_\theta) + \ln\left|\boldsymbol{\Sigma}_\theta^{-1}\right|\right] + \text{Const}, \end{aligned}$$

where $|\mathbf{A}|$ denotes the determinant of a square matrix $\mathbf{A}$. We maximize L to estimate the optimal values of $\boldsymbol{\theta}_i$, σ, $\mathbf{m}_\theta$ and $\boldsymbol{\Sigma}_\theta$. Since the number of parameters to estimate is large, we use an alternating method to solve the problem.

In the $(t+1)$st iteration, given $\mathbf{m}_\theta^{(t)}$ and $\boldsymbol{\Sigma}_\theta^{(t)}$ as estimates of $\mathbf{m}_\theta$ and $\boldsymbol{\Sigma}_\theta$ from the tth iteration, we apply gradient ascent to maximize the log-likelihood to estimate $\boldsymbol{\theta}_i^{(t+1)}$ and $\sigma^{(t+1)}$. The form of the kernel function we adopt is $k(\mathbf{x}_1, \mathbf{x}_2) = \theta_1\mathbf{x}_1^T\mathbf{x}_2 + \theta_2\exp(-\frac{\|\mathbf{x}_1-\mathbf{x}_2\|_2^2}{2\theta_3^2})$ where $\|\cdot\|_2$ denotes the 2-norm of a vector. Since each element of $\boldsymbol{\theta}_i$ and σ is positive, we instead treat $\ln\boldsymbol{\theta}_i$ and $\ln\sigma$ as variables, where each element of $\ln\boldsymbol{\theta}_i$ is the logarithm of the corresponding element in $\boldsymbol{\theta}_i$. The gradients of the log-likelihood with respect to $\ln\boldsymbol{\theta}_i$ and $\ln\sigma$ can be computed as:

$$\begin{aligned} \frac{\partial L}{\partial \ln\sigma} &= \frac{\partial L}{\partial \sigma^2}\frac{\partial \sigma^2}{\partial \ln\sigma} \\ &= \sigma^2\sum_{i=1}^{m}\left\{(\mathbf{y}^i)^T(\mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2\mathbf{I}_{n_i})^{-2}\mathbf{y}^i - \text{tr}\left[(\mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2\mathbf{I}_{n_i})^{-1}\right]\right\} \\ \frac{\partial L}{\partial \ln\boldsymbol{\theta}_i} &= \frac{1}{2}\text{diag}(\boldsymbol{\theta}_i)\left[\text{Tr}\left(\mathbf{A}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}}{\partial \boldsymbol{\theta}_i}\right) - 2(\boldsymbol{\Sigma}_\theta^{(t)})^{-1}(\boldsymbol{\theta}_i - \mathbf{m}_\theta^{(t)})\right], \end{aligned}$$

where $\mathbf{A} = (\mathbf{K}_{\boldsymbol{\theta}_i}+\sigma^2\mathbf{I}_{n_i})^{-1}\mathbf{y}^i(\mathbf{y}^i)^T(\mathbf{K}_{\boldsymbol{\theta}_i}+\sigma^2\mathbf{I}_{n_i})^{-1} - (\mathbf{K}_{\boldsymbol{\theta}_i}+\sigma^2\mathbf{I}_{n_i})^{-1}$, $\text{tr}(\cdot)$ denotes the trace function defined on a square matrix, $\text{Tr}(\mathbf{A}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}}{\partial \boldsymbol{\theta}_i})$ denotes a vector whose jth element is $\text{tr}(\mathbf{A}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}}{\partial \theta_{ij}})$ where θ_{ij} is the jth element of $\boldsymbol{\theta}_i$, and $\text{diag}(\boldsymbol{\theta}_i)$ denotes the diagonal matrix whose (j,j)th element is the jth element of $\boldsymbol{\theta}_i$.

After we obtain $\boldsymbol{\theta}_i^{(t+1)}$ and $\sigma^{(t+1)}$, we keep them fixed and maximize the log-likelihood with respect to $\mathbf{m}_\theta$ and $\boldsymbol{\Sigma}_\theta$. With some simple algebraic calculations, we can get

$$\begin{aligned} \mathbf{m}_\theta^{(t+1)} &= \frac{1}{m}\sum_{i=1}^{m}\boldsymbol{\theta}_i \\ \boldsymbol{\Sigma}_\theta^{(t+1)} &= \frac{1}{m}\sum_{i=1}^{m}(\boldsymbol{\theta}_i - \mathbf{m}_\theta^{(t+1)})(\boldsymbol{\theta}_i - \mathbf{m}_\theta^{(t+1)})^T. \end{aligned}$$

These two steps are repeated until the model parameters converge.

Given a test data point $\mathbf{x}_\star^i$ of task T_i, the predictive distribution $p(y_\star^i \mid x_\star^i, \mathbf{X}^i, \mathbf{y}^i)$ is a Gaussian distribution with mean $m_\star^i$ and variance $(\sigma_\star^i)^2$ given by

$$m_\star^i = (\mathbf{k}_\star^i)^T (\mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2 \mathbf{I}_{n_i})^{-1} \mathbf{y}^i$$
$$(\sigma_\star^i)^2 = k_{\boldsymbol{\theta}_i}(\mathbf{x}_\star^i, \mathbf{x}_\star^i) - (\mathbf{k}_\star^i)^T (\mathbf{K}_{\boldsymbol{\theta}_i} + \sigma^2 \mathbf{I}_{n_i})^{-1} \mathbf{k}_\star^i,$$

where $k_{\boldsymbol{\theta}_i}(\cdot, \cdot)$ denotes the kernel function parameterized by $\boldsymbol{\theta}_i$ and $\mathbf{k}_\star^i = (k_{\boldsymbol{\theta}_i}(\mathbf{x}_\star^i, \mathbf{x}_1^i), \ldots, k_{\boldsymbol{\theta}_i}(\mathbf{x}_\star^i, \mathbf{x}_{n_i}^i))^T$.

The computational complexity of our model is $O\left(\sum_{i=1}^m (n_i)^3\right)$. Since the data set sizes n_i for different tasks are generally small in typical semi-supervised learning applications, our model is usually quite efficient.

2.2 Inductive Inference for New Tasks

The model presented above assumes that all tasks are given in advance for multi-task learning to take place. This setting is sometimes referred to as symmetric multi-task learning [14]. If a newly arrived task does not belong to any of the tasks in the training set, our model can still deal with this situation easily without having to retrain the whole model from scratch using an augmented training set. Instead, we can utilize the common prior in Eq. (3) as the prior of the kernel parameters for the new task and then perform maximum a posteriori (MAP) estimation to obtain the kernel parameters and maximum likelihood estimation (MLE) to obtain the noise level. Therefore, we only need to store $\mathbf{m}_\theta$ and $\boldsymbol{\Sigma}_\theta$ instead of all the training data points for all tasks.

3 Semi-Supervised Multi-Task Regression

We now extend the SMTR model to the semi-supervised setting, which is called SSMTR. For task T_i, the training set D_i consists of a set of labeled data points $\{(\mathbf{x}_j^i, y_j^i)\}_{j=1}^{l_i}$ and a set of unlabeled data points $\{\mathbf{x}_j^i\}_{j=l_i+1}^{n_i}$. Typically, we have $n_i \gg l_i$.

Like in many semi-supervised learning methods which are based on the manifold assumption as described above, the unlabeled data in our model serve to enforce the smoothness of the regression function. For each task T_i, we use *local scaling* [26] to construct the similarity graph $\mathbf{S}^i$ in which each element is defined as follows:

$$S_{jr}^i = \begin{cases} \exp\left(-\frac{\|\mathbf{x}_j^i - \mathbf{x}_r^i\|^2}{\sigma_j^i \sigma_r^i}\right) & \text{if } \mathbf{x}_j^i \in N_K(\mathbf{x}_r^i) \text{ or } \mathbf{x}_r^i \in N_K(\mathbf{x}_j^i) \\ 0 & \text{otherwise} \end{cases}$$

where $N_K(\mathbf{x}_j^i)$ denotes the neighborhood set of the K nearest neighbors of $\mathbf{x}_j^i$ in task T_i, σ_j^i is the distance between $\mathbf{x}_j^i$ and its Kth nearest neighbor, and σ_r^i is the distance between $\mathbf{x}_r^i$ and its Kth nearest neighbor.

We introduce a random variable G^i to reflect the geometric structure contained in the training set of task T_i. The prior for G is defined as

$$p(G^i \mid \mathbf{f}^i, D_i) \propto \exp\left[-\frac{\alpha_i}{2}(\mathbf{f}^i)^T \mathbf{L}^i \mathbf{f}^i\right], \tag{4}$$

where $\mathbf{f}^i = (f_1^i, \ldots, f_{n_i}^i)^T$ includes the latent variables for both labeled and unlabeled data, $\mathbf{L}^i$ is the Laplacian matrix or normalized Laplacian matrix [27] of the similarity graph $\mathbf{S}^i$ defined on the training set D_i, and α_i is a hyperparameter which needs to be estimated. So if the probability of G^i is high, it means that the data set is more likely to contain manifold structure according to the graph structure implied by $\mathbf{L}^i$.

Thus the joint prior of $\mathbf{f}^i$ conditioned on D_i and G^i can be computed based on

$$p(\mathbf{f}^i \mid D_i, G^i) \propto p(\mathbf{f}^i \mid D_i) p(G^i \mid \mathbf{f}^i, D_i)$$

and so

$$\mathbf{f}^i \mid D_i, G^i \sim \mathcal{N}\left(\mathbf{0}_n, (\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \alpha_i \mathbf{L}^i)^{-1}\right). \tag{5}$$

This formulation is similar to that of [28]. However, [28] focused on semi-supervised classification but not the regression problem. The graphical model for SMTR is depicted in Figure 2.

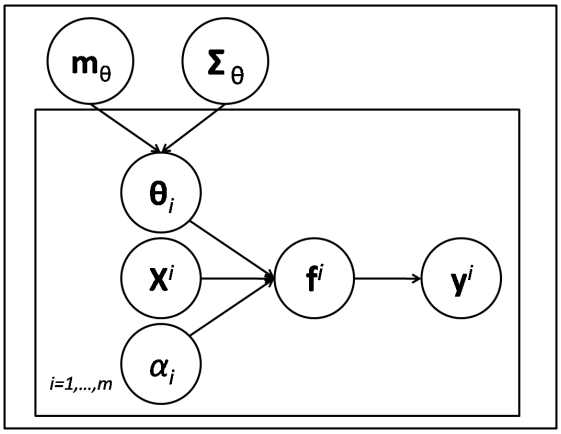

Fig. 2. Graphical model for Semi-Supervised Multi-Task Regression. Here $\mathbf{X}^i$ contains labeled and unlabeled data in the ith task.

From the joint prior defined in Eq. (5), the new kernel function for task T_i can be defined as:

$$k_i(\mathbf{x}, \mathbf{z}) = k_{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{z}) - (\mathbf{k}_{\mathbf{x}}^i)^T \left(\alpha_i^{-1}\mathbf{I} + \mathbf{L}^i \mathbf{K}_{\boldsymbol{\theta}_i}\right)^{-1} \mathbf{L}^i \mathbf{k}_{\mathbf{z}}^i, \tag{6}$$

where $\mathbf{k}_{\mathbf{x}}^i = \left(k_{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{x}_1^i), \ldots, k_{\boldsymbol{\theta}_i}(\mathbf{x}, \mathbf{x}_{n_i}^i)\right)^T$. The kernel function in Eq. (6) is similar to the semi-supervised kernel function defined in [29].

The hyperparameter α_i in Eq. (4) can be viewed as a measure of usefulness of the unlabeled data. We expect to automatically learn α_i from data. If the optimal α_i is very small or even 0 after learning, then the prior of $\mathbf{f}^i$ will degenerate to the Gaussian prior. This means that the unlabeled data points have negligible effect on improving the performance of GP regression. From the new kernel function in Eq. (6), we can view α_i as a parameter in the kernel function.

There exist some works on semi-supervised or transductive GP regression that work in a different way, such as [30]. The assumption of [30] is that the mean and variance of the predictive results on the unlabeled data are close to those of the labeled data. We can show that it is easy to incorporate this more restrictive assumption into our model.

3.1 Learning and Inference

Since the likelihood is only related to the labeled data, we first marginalize the joint prior with respect to f_j^i corresponding to the unlabled data. In this section, we still use $\mathbf{K}_{\boldsymbol{\theta}_i}$ to denote the kernel matrix whose elements are calculated by the modified kernel function in Eq. (6) on the labeled data of task T_i.

We still use an alternating method to maximize the log-likelihood. The update rules for $\mathbf{m}_\theta$, $\boldsymbol{\Sigma}_\theta$, $\boldsymbol{\theta}_i$ and σ are the same as those in Section 2.1 with the kernel function being the only difference. Moreover, for α_i, the gradient can be calculated as

$$\frac{\partial L}{\partial \ln \alpha_i} = \frac{\alpha_i}{2}\left\{\text{tr}\left[(\mathbf{K}_{\boldsymbol{\theta}_i}+\sigma^2\mathbf{I}_{l_i})^{-1}\mathbf{y}^i(\mathbf{y}^i)^T(\mathbf{K}_{\boldsymbol{\theta}_i}+\sigma^2\mathbf{I}_{l_i})^{-1}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}}{\partial \alpha_i}\right]\right.$$
$$\left.-\text{tr}\left[(\mathbf{K}_{\boldsymbol{\theta}_i}+\sigma^2\mathbf{I}_{l_i})^{-1}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}}{\partial \alpha_i}\right]\right\}.$$

When making prediction, the formulation is the same as conventional GP. Moreover, the way to handle new tasks is the same as that in Section 2.2.

4 Utilizing Pairwise Information

In the previous section, we showed that incorporating unlabeled data into SMTR to give the SSMTR model only requires modifying the GP prior, but the likelihood is still defined based solely on the labeled data.

In addition to unlabeled data, in some applications the training set also contains some other auxiliary data in the form of pairwise constraints [21]. Let the jth pairwise constraint for task T_i take the form $(i, u(j), v(j), d_j^i)$, which means that $y_{u(j)}^i - y_{v(j)}^i \geq d_j^i$ where $y_{u(j)}^i$ and $y_{v(j)}^i$ are the true outputs of two data points $\mathbf{x}_{u(j)}^i$ and $\mathbf{x}_{v(j)}^i$ in task T_i with at least one of them being an unlabeled point. For personalized pose estimation, it is easy to add a constraint that the pose angle difference between a frontal face image and a left profile face image is not less than 45 degrees.

In semi-supervised classification or clustering applications, one may also find pairwise constraints such as 'must-link' and 'cannot-link' constraints [31], which state whether or not two data points should belong to the same class or cluster. Many methods have been proposed to incorporate such pairwise constraints into their learning models. For semi-supevised regression, however, very little has been studied on this topic. Here we offer a preliminary study in the context of SSMTR.

The jth pairwise constraint of task T_i is denoted by ξ_j^i. The noise-free likelihood function $p_{\text{ideal}}(\xi_j^i \mid f_{u(j)}^i, f_{v(j)}^i)$ is defined as

$$p_{\text{ideal}}(\xi_j^i \mid f_{u(j)}^i, f_{v(j)}^i) = \begin{cases} 1 & \text{if } f_{u(j)}^i - f_{v(j)}^i \geq d_j^i \\ 0 & \text{otherwise} \end{cases}$$

In real applications, however, the pairwise constraints are often noisy. To model this more realistic setting, we introduce a random variable δ which follows some normal distribution with zero mean and unknown variance ε^2. The variance is the same for all tasks. So the corresponding likelihood function is defined as

$$\begin{aligned} & p(\xi_j^i \mid f_{u(j)}^i, f_{v(j)}^i) \\ &= \int\int p_{\text{ideal}}(\xi_j^i \mid f_{u(j)}^i + \delta_1, f_{v(j)}^i + \delta_2)\, \mathcal{N}(\delta_1 \mid 0, \varepsilon^2)\mathcal{N}(\delta_2 \mid 0, \varepsilon^2)\, d\delta_1 d\delta_2 \\ &= \Phi\left(\frac{f_{u(j)}^i - f_{v(j)}^i - d_j^i}{\sqrt{2}\varepsilon}\right), \end{aligned}$$

where $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(a \mid 0, 1)da$ is the probit function.

The noise level ε^2 in the pairwise constraints has some relationship to the noise level σ^2 in the likelihood function since they both relate the latent variable f_j^i to the output y_j^i. However, it should be noted that the noise sources they represent are different. For instance, one may have noise in the pairwise constraints but not in the likelihood, or their noise levels may be different. For flexibility, we use two different parameters for the two noise sources in our model.

Although it appears that the likelihood function of our model is similar to that of [32], their differences are worth pointing out here. The model in [32] is for classification and the constraints there refer to label preference. On the other hand, our model is for semi-supervised regression with pairwise constraints as auxiliary data and the constraints specify the differences between the outputs of pairs of data points. Moreover, the likelihood function in [32] can be seen as a special case of our likelihood function when each d_j^i takes the value 0.

4.1 Learning and Inference

Since direct integration of $\mathbf{f}^i$ is intractable, we resort to the Laplace approximation [33] to approximate the posterior of $\mathbf{f}^i$. We first compute $\mathbf{f}_{\text{MAP}}^i$ by maximizing the posterior of $\mathbf{f}^i$, which is equivalent to minimizing the following function:

$$g(\mathbf{f}^i) = \frac{1}{2}(\mathbf{f}^i)^T \mathbf{K}_{\boldsymbol{\theta}_i}^{-1}\mathbf{f}^i + \frac{\sigma^{-2}}{2}\|\tilde{\mathbf{y}}^i - \tilde{\mathbf{f}}^i\|_2^2 - \sum_{j=1}^{c_i} \ln \Phi(\omega_j^i) + l_i \ln \sigma + \frac{1}{2}\ln|\mathbf{K}_{\boldsymbol{\theta}_i}|,$$

where $\tilde{\mathbf{y}}^i$ and $\tilde{\mathbf{f}}^i$ denote the subsets of $\mathbf{y}^i$ and $\mathbf{f}^i$, respectively, corresponding to the labeled data, c_i is the number of pairwise constraints available in task T_i, and $\omega_j^i = \frac{f_{u(j)}^i - f_{v(j)}^i - d_j^i}{\sqrt{2}\varepsilon}$. We want to find $\mathbf{f}_{\text{MAP}}^i$ that minimizes $g(\mathbf{f}^i)$:

$$\mathbf{f}_{\text{MAP}}^i = \arg\min_{\mathbf{f}^i} g(\mathbf{f}^i).$$

It is easy to show that $g(\mathbf{f}^i)$ is a convex function since the Hessian matrix of $g(\mathbf{f}^i)$ is $\frac{\partial^2 g(\mathbf{f}^i)}{\partial \mathbf{f}^i \partial (\mathbf{f}^i)^T} = \mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}^i$, which is positive definite, where $\mathbf{I}_{n_i}^{l_i}$ is the $n_i \times n_i$

zero matrix with the first l_i diagonal elements being 1, and $\boldsymbol{\Omega}^i = \frac{\partial^2 - \sum_{j=1}^{c_i} \ln \Phi(\omega_j^i)}{\partial \mathbf{f}^i \partial (\mathbf{f}^i)^T}$ is positive semidefinite. The proof is similar to that in [32] and we omit it here. So we can apply gradient descent to find the global optimum.

After obtaining $\mathbf{f}_{\text{MAP}}^i$, we can approximate the likelihood or evidence of task T_i according to the analysis in [33] as

$$p(\mathbf{y}^i) \approx \exp\{-g(\mathbf{f}_{\text{MAP}}^i)\} \frac{(2\pi)^{n_i/2}}{|\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}_{\text{MAP}}^i|^{1/2}},$$

where $\boldsymbol{\Omega}_{\text{MAP}}^i$ is the value of the function $\boldsymbol{\Omega}^i$ taking on $\mathbf{f}_{\text{MAP}}^i$.

So the total negative log-likelihood of all m tasks can be approximated as

$$\begin{aligned} L = & \sum_{i=1}^{m} \Big[\frac{1}{2} (\mathbf{f}_{\text{MAP}}^i)^T \mathbf{K}_{\boldsymbol{\theta}_i}^{-1} \mathbf{f}_{\text{MAP}}^i + \frac{\sigma^{-2}}{2} \|\tilde{\mathbf{y}}^i - \tilde{\mathbf{f}}_{\text{MAP}}^i\|_2^2 + \\ & \qquad \frac{1}{2} \ln |\mathbf{K}_{\boldsymbol{\theta}_i}| + \frac{1}{2} \ln |\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}_{\text{MAP}}^i| \Big] + \\ & \sum_{i=1}^{m} \frac{1}{2} \Big[(\boldsymbol{\theta}_i - \mathbf{m}_\theta)^T \boldsymbol{\Sigma}_\theta^{-1} (\boldsymbol{\theta}_i - \mathbf{m}_\theta) + \ln |\boldsymbol{\Sigma}_\theta^{-1}| \Big] + \\ & \sum_{i=1}^{m} \Big[l_i \ln \sigma - \sum_{j=1}^{c_i} \ln \Phi(\omega_j^i) \Big] + \text{Const.} \end{aligned}$$

We still use an alternating method to minimize the negative log-likelihood. In the $(t+1)$st iteration, given $\mathbf{m}_\theta^{(t)}$ and $\boldsymbol{\Sigma}_\theta^{(t)}$, the gradient of L with respect to each variable is given by

$$\begin{aligned} \frac{\partial L}{\partial \ln \sigma} &= \sigma^2 \sum_{i=1}^{m} \Big\{ -\sigma^{-4} \|\tilde{\mathbf{y}}^i - \tilde{\mathbf{f}}_{\text{MAP}}^i\|_2^2 + \frac{l_i}{\sigma^2} - \sigma^{-4} \text{tr} \left[(\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}_{\text{MAP}}^i)^{-1} \right] \Big\} \\ \frac{\partial L}{\partial \ln \boldsymbol{\theta}_i} &= \frac{\text{diag}(\boldsymbol{\theta}_i)}{2} \left\{ \text{Tr}(\mathbf{B} \frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}^{-1}}{\partial \boldsymbol{\theta}_i}) + 2(\boldsymbol{\Sigma}_\theta^{(t)})^{-1} (\boldsymbol{\theta}_i - \mathbf{m}_\theta^{(t)}) \right\} \\ \frac{\partial L}{\partial \ln \alpha_i} &= \frac{\alpha_i}{2} \left\{ -(\mathbf{f}_{\text{MAP}}^i)^T \frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}^{-1}}{\partial \alpha_i} \mathbf{f}_{\text{MAP}}^i - \text{tr} \left(\mathbf{K}_{\boldsymbol{\theta}_i} \frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}^{-1}}{\partial \alpha_i} \right) \right. \\ &\qquad \left. + \text{tr} \left[(\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}_{\text{MAP}}^i)^{-1} \frac{\partial \mathbf{K}_{\boldsymbol{\theta}_i}^{-1}}{\partial \alpha_i} \right] \right\} \\ \frac{\partial L}{\partial \ln \varepsilon} &= \varepsilon \sum_{i=1}^{m} \left\{ \frac{1}{2} \text{tr} \left[(\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}_{\text{MAP}}^i)^{-1} \frac{\partial \boldsymbol{\Omega}_{\text{MAP}}^i}{\partial \varepsilon} \right] + \sum_{j=1}^{c_i} \frac{\omega_j^i \mathcal{N}(\omega_j^i | 0, 1)}{\varepsilon \Phi(\omega_j^i)} \right\}, \end{aligned}$$

where $\mathbf{B} = \mathbf{f}_{\text{MAP}}^i (\mathbf{f}_{\text{MAP}}^i)^T - \mathbf{K}_{\boldsymbol{\theta}_i} + \left(\mathbf{K}_{\boldsymbol{\theta}_i}^{-1} + \sigma^{-2}\mathbf{I}_{n_i}^{l_i} + \boldsymbol{\Omega}_{\text{MAP}}^i \right)^{-1}$.

The update rules for $\mathbf{m}_\theta$ and $\boldsymbol{\Sigma}_\theta$ are the same as those in Section 2.1.

4.2 Extension

In some applications, there exist auxiliary data given in another form as $\xi_j^i = (i, u(j), v(j), w(j), z(j))$, which means that $y_{u(j)}^i - y_{v(j)}^i \geq y_{w(j)}^i - y_{z(j)}^i$. Let us take the personalized pose estimation problem again as example. It is often easy to know that the pose angle difference between a left profile face image and a right profile face image is larger than that of two nearly frontal face images. The pairwise constraints considered before may be seen as a special case of ξ_j^i when two of the data points $\mathbf{x}_{u(j)}^i, \mathbf{x}_{v(j)}^i, \mathbf{x}_{w(j)}^i$ and $\mathbf{x}_{z(j)}^i$ are labeled. The special case with $v(j) = w(j)$ is also interesting in the pose estimation application. For example, the pose angle difference between a left profile face image and a right profile face image is larger than that between the left profile image and a frontal image. Similar to the pairwise constraints above, the noise-free likelihood function $p_{\text{ideal}}(\xi_j^i \mid f_{u(j)}^i, f_{v(j)}^i, f_{w(j)}^i, f_{z(j)}^i)$ is defined as

$$p_{\text{ideal}}(\xi_j^i \mid f_{u(j)}^i, f_{v(j)}^i, f_{w(j)}^i, f_{z(j)}^i) = \begin{cases} 1 & \text{if } f_{u(j)}^i - f_{v(j)}^i \geq f_{w(j)}^i - f_{z(j)}^i \\ 0 & \text{otherwise} \end{cases}$$

For more realistic situations, we again introduce a random variable δ following a normal distribution with zero mean and unknown variance ε^2. The likelihood function is thus defined as

$$\begin{aligned} & p(\xi_j^i \mid f_{u(j)}^i, f_{v(j)}^i, f_{w(j)}^i, f_{z(j)}^i) \\ & = \int p_{\text{ideal}}(\xi_j^i \mid f_{u(j)}^i + \delta_1, f_{v(j)}^i + \delta_2, f_{w(j)}^i + \delta_3, f_{z(j)}^i + \delta_4) \\ & \mathcal{N}(\delta_1 \mid 0, \varepsilon^2)\mathcal{N}(\delta_2 \mid 0, \varepsilon^2)\mathcal{N}(\delta_3 \mid 0, \varepsilon^2)\mathcal{N}(\delta_4 \mid 0, \varepsilon^2)\, d\boldsymbol{\delta} \\ & = \Phi\left(\frac{f_{u(j)}^i - f_{v(j)}^i - f_{w(j)}^i + f_{z(j)}^i}{2\varepsilon}\right), \end{aligned}$$

where $\boldsymbol{\delta} = (\delta_1, \delta_2, \delta_3, \delta_4)^T$. It is easy to show that the posterior distribution of $\mathbf{f}^i$ is unimode, so we can also use the Laplace approximation to make inference.

5 Experiments

We compare a single-task regression method, SMTR, SSMTR and SSMTR with pairwise information in this section using two benchmark data sets, one for learning the inverse dynamics of a robot arm and another for predicting the student performance in terms of examination scores.

5.1 Learning Inverse Dynamics

This data set[1] was used in [34]. For each instance, there are 7 joint positions, 7 joint velocities and 7 joint accelerations forming 21 input attributes, together with 7 joint torques for the outputs corresponding to 7 degrees of freedom. We treat each output

[1] http://www.gaussianprocess.org/gpml/data/

(i.e., degree of freedom) as a separate learning task. To simulate a more general multi-task learning setting, we randomly select 2000 data points independently for each task so that the input data points for different tasks are different. We randomly partition the whole data set into three subsets, with 1% as labeled data, 10% as unlabeled data and the rest as test data. The kernel we use is the RBF kernel. Moreover, we randomly select 100 pairs of data points and generate the pairwise constraints using their labels. Ten random splits are performed and the mean and standard derivation of the performance measure over different splits are reported. We adopt the normalized mean squared error (nMSE), which is defined as the mean squared error divided by the variance of the test label, as the performance measure. Table 2 shows the results. From the results, we can see that the performance of our proposed SMTR is significantly better than that of supervised single-task learning which uses one GP for each task. Moreover, the performance of SSMTR and SSMTR using pairwise information is better than that of SMTR, which shows that both the unlabeled data and the pairwise information are effective in improving performance.

Table 1. nMSE results on learning inverse dynamics (SSTR: supervised single-task regression which uses one GP for each task; SSMTRPI: semi-supervised multi-task regression with pairwise information).

Method	Transductive Error	Inductive Error
SSTR	1.0228±0.1318	1.0270±0.1450
SMTR	0.4149±0.1109	0.4368±0.1020
SSMTR	0.3810±0.1080	0.3905±0.1123
SSMTRPI	0.3500±0.1088	0.3486±0.1010

5.2 Predicting Student Performance

This data set[2] was used in [11] for multi-task learning. The goal is to predict the student performance in terms of examination scores. The data set consists of 15362 students from 139 secondary schools, recording their examination scores in three years (1985–87). We treat each school as a different task, so that there are 139 learning tasks in total. For each instance, the input consists of the year of the examination as well as 4 school-specific and 3 student-specific attributes. We still use nMSE as performance measure. We randomly select 2% of the data as labeled data, 20% as unlabeled data, and the rest as test data. The kernel we adopt is the RBF kernel. We also generate 100 pairwise constraints just as the last experiment did. We perform 10 random splits and report the mean and standard derivation over different splits. Table 2 shows the results. Similar to the results on learning inverse dynamics, SSMTR with pairwise information gives the best performance.

[2] http://www.cs.ucl.ac.uk/staff/A.Argyriou/code/

Table 2. nMSE results on predicting student performance

Method	Transductive Error	Inductive Error
SSTR	1.2914±0.3146	1.3240±0.3274
SMTR	1.1151±0.3025	1.1535±0.3128
SSMTR	1.0506±0.2804	1.0612±0.2813
SSMTRPI	0.9817±0.2809	0.9824±0.2832

6 Conclusion

In this paper, we have proposed an approach for integrating semi-supervised regression and multi-task regression under a common framework. We first propose a new supervised multi-task regression method based on GP and then extend it to incorporate unlabeled data by modifying the GP prior. In addition, if auxiliary data in the form of pairwise constraints are available, we propose a scheme to incorporate them into our semi-supervised multi-task regression framework by modifying the likelihood term. In our future research, we will investigate sparse extension of our models, possibly by using the informative vector machine [35].

Acknowledgments

This research has been supported by General Research Fund 621407 from the Research Grants Council of the Hong Kong Special Administrative Region, China.

References

1. Chapelle, O., Zien, A., Schölkopf, B., eds.: Semi-Supervised Learning. MIT Press, Boston (2006)
2. Blum, A., Mitchell, T.: Combining labeled and unlabeled data with co-training. In: Proceedings of the Workshop on Computational Learning Theory, Madison, Wisconsin, USA (1998) 92–100
3. Bennett, K., Demiriz, A.: Semi-supervised support vector machines. In: Advances in Neural Information Processing Systems 11, Vancouver, British Columbia, Canada (1998) 368–374
4. Joachims, T.: Transductive inference for text classification using support vector machines. In: Proceedings of the Sixteenth International Conference on Machine Learning, San Francisco, CA, USA (1999) 200–209
5. Zhou, D., Bousquet, O., Lal, T., Weston, J., Schölkopf, B.: Learning with local and global consistency. In: Advances in Neural Information Processing Systems 16, Vancouver, British Columbia, Canada (2003)
6. Zhu, X., Ghahramani, Z., Lafferty, J.: Semi-supervised learning using Gaussian fields and harmonic functions. In: Proceedings of the Twentieth International Conference on Machine Learning, Washington, DC (2003) 912–919
7. Belkin, M., Niyogi, P., Sindhwani, V.: Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. Journal of Machine Learning Research **7** (2006) 2399–2434
8. Caruana, R.: Multitask learning. Machine Learning **28**(1) (1997) 41–75

9. Baxter, J.: A Bayesian/information theoretic model of learning to learn via multiple task sampling. Machine Learning **28**(1) (1997) 7–39
10. Thrun, S.: Is learning the n-th thing any easier than learning the first? In Touretzky, D.S., Mozer, M., Hasselmo, M.E., eds.: Advances in Neural Information Processing Systems 8, Denver, CO (1996) 640–646
11. Argyriou, A., Evgeniou, T., Pontil, M.: Convex multi-task feature learning. Machine Learning **73**(3) (2008) 243–272
12. Evgeniou, T., Pontil, M.: Regularized multi-task learning. In: Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, Seattle, Washington, USA (2004) 109–117
13. Bakker, B., Heskes, T.: Task clustering and gating for Bayesian multitask learning. Journal of Machine Learning Research **4** (2003) 83–99
14. Xue, Y., Liao, X., Carin, L., Krishnapuram, B.: Multi-task learning for classification with Dirichlet process priors. Journal of Machine Learning Research **8** (2007) 35–63
15. Lawrence, N.D., Platt, J.C.: Learning to learn with the informative vector machine. In: Proceedings of the Twenty-first International Conference on Machine Learning, Banff, Alberta, Canada (2004)
16. Schwaighofer, A., Tresp, V., Yu, K.: Learning Gaussian process kernels via hierarchical Bayes. In Saul, L.K., Weiss, Y., Bottou, L., eds.: Advances in Neural Information Processing Systems 17, Vancouver, British Columbia, Canada (2005) 1209–1216
17. Yu, K., Tresp, V., Schwaighofer, A.: Learning Gaussian processes from multiple tasks. In: Proceedings of the Twenty-Second International Conference on Machine Learning, Bonn, Germany (2005) 1012–1019
18. Bonilla, E., Chai, K.M.A., Williams, C.: Multi-task Gaussian process prediction. In Platt, J., Koller, D., Singer, Y., Roweis, S., eds.: Advances in Neural Information Processing Systems 20, Vancouver, British Columbia, Canada (2008) 153–160
19. Ando, R.K., Zhang, T.: A framework for learning predictive structures from multiple tasks and unlabeled data. Journal of Machine Learning Research **6** (2005) 1817–1853
20. Liu, Q., Liao, X., Carin, L.: Semi-supervised multitask learning. In Platt, J., Koller, D., Singer, Y., Roweis, S., eds.: Advances in Neural Information Processing Systems 20, Vancouver, British Columbia, Canada (2008) 937–944
21. Zhu, X., Goldberg, A.: Kernel regression with order preferences. In: Proceedings of the Twenty-Second AAAI Conference on Artificial Intelligence, Vancouver, British Columbia, Canada (2007) 681–686
22. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning. The MIT Press, Cambridge, MA, USA (2006)
23. Yu, S., Tresp, V., Yu, K.: Robust multi-task learning with t-processes. In: Proceedings of the Twenty-Fourth International Conference on Machine Learning, Corvalis, Oregon, USA (2007) 1103–1110
24. Heskes, T.: Solving a huge number of similar tasks: a combination of multi-task learning and a hierarchical Bayesian approach. In: Proceedings of the Fifteenth International Conference on Machine Learning, Madison, Wisconson, USA (1998) 233–241
25. Heskes, T.: Empirical bayes for learning to learn. In: Proceedings of the Seventeenth International Conference on Machine Learning, Stanford University, Standord, CA, USA (2000) 367–374
26. Zelnik-Manor, L., Perona, P.: Self-tuning spectral clustering. In: Advances in Neural Information Processing Systems 17, Vancouver, British Columbia, Canada (2005) 1601–1608
27. Chung, F.R.K.: Spectral Graph Theory. American Mathematical Society, Rhode Island (1997)

28. Sindhwani, V., Chu, W., Keerthi, S.S.: Semi-supervised Gaussian process classifiers. In: Proceedings of the 20th International Joint Conference on Artificial Intelligence, Hyderabad, India (2007) 1059–1064
29. Sindhwani, V., Niyogi, P., Belkin, M.: Beyond the point cloud: from transductive to semi-supervised learning. In: Proceedings of the Twenty-Second International Conference on Machine Learning, Bonn, Germany (2005) 824–831
30. Le, Q.V., Smola, A.J., Gärtner, T., Altun, Y.: Transductive Gaussian process regression with automatic model selection. In: Proceedings of the 17th European Conference on Machine Learning, Berlin, Germany (2006) 306–317
31. Bilenko, M., Basu, S., Mooney, R.J.: Integrating constraints and metric learning in semi-supervised clustering. In: Proceedings of the Twenty-first International Conference on Machine Learning, Banff, Alberta, Canada (2004)
32. Chu, W., Ghahramani, Z.: Preference learning with Gaussian processes. In: Proceedings of the Twenty-Second International Conference on Machine Learning. (2005) 137–144
33. Bishop, C.M.: Pattern Recognition and Machine Learning. Springer, New York (2006)
34. Vijayakumar, S., D'Souza, A., Schaal, S.: Incremental online learning in high dimensions. Neural Computation **17**(12) (2005) 2602–2634
35. Lawrence, N.D., Seeger, M., Herbrich, R.: Fast sparse Gaussian process methods: The informative vector machine. In Becker, S., Thrun, S., Obermayer, K., eds.: Advances in Neural Information Processing Systems 15, Vancouver, British Columbia, Canada (2003) 609–616