Learning Gaussian Processes from Multiple Tasks

Kai Yu KAI.YU@SIEMENS.COM
Information and Communication, Corporate Technology, Siemens AG, Munich, Germany

Volker Tresp VOLKER.TRESP@SIEMENS.COM
Information and Communication, Corporate Technology, Siemens AG, Munich, Germany

Anton Schwaighofer ANTON@FIRST.FHG.DE
Intelligent Data Analysis Group, Fraunhofer FIRST, Berlin

Abstract

We consider the problem of multi-task learning, that is, learning multiple related functions. Our approach is based on a hierarchical Bayesian framework, that exploits the equivalence between parametric linear models and nonparametric Gaussian processes (GPs). The resulting models can be learned easily via an EM-algorithm. Empirical studies on multi-label text categorization suggest that the presented models allow accurate solutions of these multi-task problems.

1. Introduction

In this paper we consider the case where there are multiple related predictive functions to estimate. Hierarchical Bayesian modeling provides a natural way of obtaining a joint regularization for individual models by assuming that model parameters are drawn from a common hyperprior. In its simplest form, hierarchical Bayesian modeling means to learn point estimates for the hyperparameters in the model. Effectively, an "informed prior" distribution is learned.

Previous efforts were mostly put into hierarchical modeling with parametric functions. Only few authors have addressed hierarchical Bayesian modeling from a nonparametric perspective. In this paper we study the application of hierarchical Bayesian modeling to Gaussian processes. First, we motivate the use of a normal-inverse Wishart prior distribution for the mean and covariance function at the training data points. Then we show that, in a setting with a fixed number of data points, we can derive a finite representation and an EM algorithm to find MAP estimates of functional values, as well as prior mean and covariance matrix. In addition, we show that also in a general inductive setting, it is possible to write down both the finite representations of the Gaussian process and EM equations. Thus, we obtain exact predictions at arbitrary test points. As a particular feature of the presented approach, it is possible to explicitly write down the "learned kernel" that captures all information extracted from the individual tasks. Moreover, as a practical issue of the suggested work, our nonparametric model is more efficient than parametric models in the situation of learning high dimensional functions, since the complexity only depends on the training size.

The paper is organized as follows. In the following section we describe the link between parametric linear models and Gaussian processes. In Sec. 3 we briefly introduce multi-task learning from a hierarchical Bayesian point of view, followed by a discussion of hierarchical linear models in Sec. 4. The main results of our paper, namely hierarchical Bayesian modeling applied to Gaussian processes, are presented in Sec. 5. Sec. 6 contains experimental results validating our theoretical analysis.

1.1. Related Work

Research on modeling multiple related functions has been carried in several strands. Multi-task learning (Caruana, 1997) is aiming at sharing knowledge gained in individual scenarios, by, for example, sharing hidden units in neural networks. A recent work is that of (Evegniou & Pontil, 2004), where a set of linear functions is used in a support vector machine framework. The model considers only the mean of those linear functions. Ando and Zhang (2004) present an iterative algorithm that alternatively estimates the weights of

Appearing in *Proceedings of the 22nd International Conference on Machine Learning*, Bonn, Germany, 2005.

linear functions and then performs PCA on the multiple functions' weights. This essentially models the covariance of the linear functions, and restricts the freedom of the common structure by the chosen dimensionality of PCA. In contrast, the model we will present subsequently considers both mean and covariance of the multiple functions. Also, the dimensionality of the common structure does not need to be chosen explicitly.

In statistics, modeling data from related scenarios is typically done via mixed effects models or hierarchical Bayesian (HB) modeling (Gelman et al., 1995). In HB, parameters of models for individual scenarios are assumed to be drawn from a common (hyper)prior distribution, allowing the individual models to interact and regularize each other. Recent examples of HB modeling in machine learning include (Bakker & Heskes, 2003; Blei et al., 2003). (Gelman et al., 1995) also lists a number of references on hierarichal Bayesian modeling, including hierarchical linear models. Most recently, Chapelle and Harchaoui (2005) applied hierarchical linear models for conjoint analysis, which appears to be similar to our models in linear case.

The work of (Lawrence & Platt, 2004) presents a multi-task approach specifically for Gaussian process models, by inferring parameters of a shared covariance function. As this is computationally very intensive, a sparse approximation scheme has to be adopted. Subsequently, (Schwaighofer et al., 2005) presented an approach to learning covariance matrices from multi-task data via an EM-algorithm. Yet, generalization to new data could only be achieved by an ad-hoc form of kernel extrapolation. In contrast, our models presented here address all these issues in one framework.

2. From Linear Models to Gaussian Processes

In the subsequent sections, we will present a number of algorithms that can be used to learn from data in multiple related scenarios. All of these algorithms will make use of the duality between parametric linear models and the equivalent Gaussian process. Thus, we will first briefly outline this equivalence. The exposition here follows (Williams, 1998).

Data are given as input/output pairs $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$, with feature vector $\mathbf{x}_i \in \mathbb{R}^d$ and the output $y_i \in \mathbb{R}$. If we assume that y is generated from a linear function $f(\mathbf{x}) = \mathbf{w}^\intercal \mathbf{x}$ with additive noise ε, the *ridge regression* estimate of f is

$$\hat{\mathbf{w}} = \arg\min_{\mathbf{w}} \sum_{i=1}^{n} (\mathbf{w}^\intercal \mathbf{x}_i - y_i)^2 + \lambda \|\mathbf{w}\|^2, \qquad (1)$$

where $\|\mathbf{w}\| = \mathbf{w}^\intercal \mathbf{w}$ is the regularizer to constrain the freedom of weights $\mathbf{w}$.

From a Bayesian point of view, the regularizer is the outcome of a prior distribution on function weights, which in this case is assumed to be a Gaussian distribution, i.e. $\mathbf{w} \sim \mathcal{N}(0, \mathbf{I})$. Given the training examples $\mathbf{D}$, the *a posteriori* distribution of $\mathbf{w}$ can be calculated by applying Bayes rule,

$$p(\mathbf{w}|\mathbf{D}) = \frac{1}{Z} p(\mathbf{y}|\mathbf{w}, \mathbf{X}) p(\mathbf{w}) \propto \exp\{-\frac{1}{2} J(\mathbf{w})\},$$

where Z is a constant independent of $\mathbf{w}$, and $J(\mathbf{w}) = \frac{1}{\sigma^2} \sum_i (\mathbf{w}^\intercal \mathbf{x}_i - y_i)^2 + \|\mathbf{w}\|^2$. It is easy to see that $J(\mathbf{w})$ is the same as the cost function in ridge regression if $\sigma^2 = \lambda$, thus the maximum *a posteriori* (MAP) estimate of $\mathbf{w}$ gives the same result as ridge regression.

Furthermore, the sufficient statistics of $p(\mathbf{w})$ completely specify the properties of the considered function space, which in turn are characterized by the *mean function*

$$\mathrm{E}(f(\mathbf{x})) = \mathrm{E}(\mathbf{w}^\intercal)\mathbf{x} = 0 \qquad (2)$$

and the *covariance function*

$$\mathrm{E}(f(\mathbf{x}_i), f(\mathbf{x}_j)) = \mathbf{x}_i^\intercal \mathbf{C}_w \mathbf{x}_j = \langle \mathbf{x}_i, \mathbf{x}_j \rangle \qquad (3)$$

where $\mathbf{C}_w = \mathbf{I}$ is the covariance matrix of $\mathbf{w}$. It is easy to see that, given any finite set $\{\mathbf{x}_i\}$, the joint distribution of $\{f(\mathbf{x}_i)\}$ is a multivariate Gaussian distribution $\mathcal{N}(0, \mathbf{K})$ with $\mathbf{K}_{i,j} = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$. Therefore, $p(\mathbf{w})$ directly specifies a *Gaussian process* (GP) prior on the function space, with a zero mean function, Eq. (2), and covariance function $K(\cdot, \cdot)$ given by Eq. (3). By appealing to the standard equations for prediction with Gaussian process, the predictive mean for training data $\mathbf{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ is given by

$$\hat{f}(\mathbf{x}) = \sum_i \alpha_i K(\mathbf{x}_i, \mathbf{x}).$$

The coefficients $\boldsymbol{\alpha} = (\mathbf{K} + \sigma^2 \mathbf{I})^{-1} \mathbf{y}$, with $\mathbf{K}$ being the covariance matrix on training points, and $\mathbf{y} = [y_1, \ldots, y_n]^\intercal$.

Viewing linear models as Gaussian processes offers a number of advantages. Firstly, GPs directly work on the kernel matrix for finite training points. Thus, the modeling complexity is independent of the dimensionality of $\mathbf{x}$, which allows us to work on high or

even infinite dimensional input spaces. Second, nonlinear functions can be handled by adopting a kernel $K(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}'_i, \mathbf{x}'_j \rangle$ via a nonlinear feature mapping $\mathbf{x}' = \phi(\mathbf{x})$.

3. Learning from Multiple Tasks

In the previous section, we have outlined linear models for the case of estimating *one* function f that is underlying some data $\mathbf{D}$. In contrast, we consider here the estimation of m *related functions* f_l, $l = 1, \ldots, m$, based on training data $\mathbf{D}_l = (\mathbf{X}_l, \mathbf{y}_l)$. We assume $\mathbf{X}_l \in \mathbb{R}^{n_l \times d}$, $\mathbf{y}_l \in \mathbb{R}^{n_l}$ and n_l is the size of training data for f_l. Since each function has a different set of labeled points, there are in total n distinct data points in $\{\mathbf{D}_l\}$ with $\min(\{n_l\}) \leq n \leq \sum_l n_l$. $\cup \mathbf{X}_l$ denotes the set of distinguished $\mathbf{x}$ in $\{\mathbf{D}_l\}$.

To allow the functions to share some common structure, one way would be to assume that $\{f_l\}$ are all sampled from a common prior $p(f)$. Thus, when trying to capture the dependency between $\{f_l\}$, one inevitably has to learn the prior $p(f)$. In general, it is very difficult to directly deal with an infinite-dimensional distribution $p(f)$, unless it is conditioned on a finite set of parameters of the form $p(f|\theta)$. In the latter case, solutions are often quite straightforward. For example, one can specify a GP prior with the covariance function as a convex combination of some given kernel functions, and then try to optimize the coefficients. In essence, this approach is taken by (Lawrence & Platt, 2004). The *maximum-likelihood* (ML) estimate of θ can be obtained by maximizing

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

Alternatively, one can also derive the *maximum penalized likelihood* estimate by specifying a hyper prior distribution $p(\theta)$ and then maximizing $p(\{\mathbf{y}_l\}|\{\mathbf{X}_l\}, \theta)p(\theta)$.

4. Multi-Task Linear Models

In this section we will consider multi-task learning with (parametric) linear models. Linear models are both highly relevant and popular for many real world applications (Zhang & Oles, 2001). More importantly, like the connection between ridge regression and GPs, linear models offer insights to nonparametric modeling and thus pave the way for our further discussions.

Instead of the fixed $p(\mathbf{w})$ we had assumed in Sec. 2 for Bayesian ridge regression, we assume $\mathbf{w} \sim \mathcal{N}(\boldsymbol{\mu}_w, \mathbf{C}_w)$ and try to estimate $\boldsymbol{\mu}_w, \mathbf{C}_w$ from the data. To obtain a maximum penalized likelihood estimate of $\boldsymbol{\mu}_w, \mathbf{C}_w$, we assume a normal-inverse-Wishart distribution as the hyper prior,

$$p(\boldsymbol{\mu}_w, \mathbf{C}_w) = \mathcal{N}(\boldsymbol{\mu}_w|\boldsymbol{\mu}_{w_0}, \frac{1}{\pi}\mathbf{C}_w)\mathcal{IW}(\mathbf{C}_w|\tau, \mathbf{C}_{w_0}). \quad (4)$$

This distribution is the conjugate prior for a multivariate Gaussian distribution (Gelman et al., 1995). It can be specified[1] by means of scale matrix $\mathbf{C}_{w_0}$ with precision (or "equivalent sample size") τ, and a prior mean $\boldsymbol{\mu}_{w_0}$ for $\boldsymbol{\mu}_w$ with precision π. Similar to the setting in Bayesian ridge regression, we assume $\mathbf{C}_{w_0} = \mathbf{I}$ and $\boldsymbol{\mu}_{w_0} = 0$. With these parameters, we obtain the following model:

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

1. *$\boldsymbol{\mu}_w, \mathbf{C}_w$ are sampled once from the hyper prior as in Eq. (4);*
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 $\epsilon \sim \mathcal{N}(0, \sigma^2)$.*

Learning such a model can be done by considering the penalized likelihood, and optimizing with respect to $\theta = \{\boldsymbol{\mu}_w, \mathbf{C}_w, \sigma^2\}$ via the following EM algorithm.[2]

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

$$\hat{\mathbf{w}}_l = \mathbf{C}_{w_l}\Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal \mathbf{y}_l + \mathbf{C}_w^{-1}\boldsymbol{\mu}_w\Big) \quad (5)$$

$$\mathbf{C}_{w_l} = \Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal \mathbf{X}_l + \mathbf{C}_w^{-1}\Big)^{-1} \quad (6)$$

- *M-step*: Optimize θ based on the last E-step.

$$\boldsymbol{\mu}_w = \frac{1}{\pi + m}\sum_l \hat{\mathbf{w}}_l \quad (7)$$

$$\mathbf{C}_w = \frac{1}{\tau + m}\Big\{\pi\boldsymbol{\mu}_w\boldsymbol{\mu}_w^\intercal + \tau\mathbf{I} + \sum_l \mathbf{C}_{w_l} + \sum_l \big[\hat{\mathbf{w}}_l - \boldsymbol{\mu}_w\big]\big[\hat{\mathbf{w}}_l - \boldsymbol{\mu}_w\big]^\intercal\Big\} \quad (8)$$

$$\sigma^2 = \frac{1}{\sum_l n_l}\sum_l \|\mathbf{y}_l - \mathbf{X}_l\hat{\mathbf{w}}_l\|^2 + \mathrm{tr}[\mathbf{X}_l\mathbf{C}_{w_l}\mathbf{X}_l^\intercal] \quad (9)$$

  Here, $\mathrm{tr}[\cdot]$ denotes the trace of a matrix.

Most computational complexity is in the E-step, where a (modified) ridge regression problem has to be solved for each of the m tasks. Thus the total cost is $O(kmd^3)$, where k is the number of EM iterations.

[1] In the literature, a number of different parameterizations of the Wishart distribution are used, yet all can be reduced to the one used here.

[2] A derivation is given in Sec. A. expectation-maximization (EM) algorithm:

5. Multi-Task Gaussian Processes

Similar to ridge regression, the learned prior $\mathcal{N}(\boldsymbol{\mu}_w, \mathbf{C}_w)$ in Model 1 defines a Gaussian process with mean function $\boldsymbol{\mu}(\mathbf{x}) = \boldsymbol{\mu}_w^\intercal \mathbf{x}$ and covariance function $K(\mathbf{x}_i, \mathbf{x}_j) = \mathbf{x}_i^\intercal \mathbf{C}_w \mathbf{x}_j$. When $\mathbf{x}$ is high-dimensional, it is usually impossible to handle the prior $\mathcal{N}(\boldsymbol{\mu}_w, \mathbf{C}_w)$ directly. For such cases, the GP view of the linear model can provide a feasible solution. Similar to the discussion in Sec. 2, we will show in the following that the multi-task linear Model 1 also has a nonparametric counterpart in the GP framework.

We assume that the feature space is sufficiently high-dimensional (or infinite), such that for the considered range of random sample size $n \ll d$, the inner product $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \mathbf{x}_i, \mathbf{x}_j \rangle$ gives a valid positive definite kernel. In case $\mathbf{x}$ is the outcome of a nonlinear mapping $\phi(\mathbf{t})$ from some original feature $\mathbf{t}$, $\kappa(\cdot, \cdot)$ actually defines a nonlinear kernel on the original feature space.

We will later refer to κ as the *base kernel* that describes the properties of the shared Wishart prior. Note that this base kernel is different from the GP kernel $K(\mathbf{x}_i, \mathbf{x}_j)$, which denotes covariances in function space.

5.1. Transductive Multi-Task GPs

Sec. 4 described a hierarchical model, by specifying properties of the weights of linear models. The following theorem now relates these to the properties of the mean and covariance function of the equivalent Gaussian process.[3]

Theorem 5.1. *Let $\mathcal{S} \subset \mathbb{R}^d$ be the set of data points, such that $\forall \mathbf{x}, \mathbf{x}' \in \mathcal{S}$, $\kappa(\mathbf{x}, \mathbf{x}') = \langle \mathbf{x}, \mathbf{x}' \rangle$ defines a positive definite kernel. Then for* any *given finite 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,*

$$p(\boldsymbol{\mu}_\mathbf{f}, \mathbf{K}) = \mathcal{N}(\boldsymbol{\mu}_\mathbf{f} | 0, \frac{1}{\pi}\mathbf{K}) \mathcal{IW}(\mathbf{K} | \tau, \boldsymbol{\kappa}), \qquad (10)$$

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

The theorem essentially states that the GP kernel matrix $\mathbf{K}$ on any subset of data points in $\mathcal{S}$ is a random sample drawn from an inverse-Wishart distribution, with the scale matrix equal to the base kernel matrix $\boldsymbol{\kappa}$ and precision τ. One can simply generalize the conclusion to the case $d \to \infty$ such that $\mathcal{S}$ can be the whole infinite-dimensional feature space. Then for any finite set of data points, the corresponding $\boldsymbol{\mu}_\mathbf{f}$ and $\mathbf{K}$ follow an inverse-Wishart distribution associated with a positive definite base kernel function $\kappa(\cdot, \cdot)$. In particular, this holds if a nonlinear mapping is used for the base kernel κ.

In this section, we restrict attention to the case that one is only interested in function values and the kernel matrix on a finite set of data (sometimes referred to as transduction or working with partially labeled data). In this case, the above theorem suggests the following generative model:

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 Eq. (10), 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 $\epsilon \sim \mathcal{N}(0, \sigma^2)$.*

The model is actually a realization of Model 1 but focusing on a finite set $\mathbf{X}$. It indicates that the joint distribution of function values on $\mathbf{X}$ is a Gaussian distribution with the hyper prior, Eq. (4), where the scale matrix is the base kernel $\kappa(\cdot, \cdot)$ evaluated on $\mathbf{X}$.

A similar model has been derived by (Schwaighofer et al., 2005). Note that Model 2 still works if $\mathbf{X}$ is expanded by including *any* number of new test points $\mathbf{x}$, as long as $\cup \mathbf{X}_l \subseteq \mathbf{X}$. This point was not well clarified in (Schwaighofer et al., 2005). We call Model 2 *transductive* since the test points must be known before the training.

5.2. Inductive Multi-Task GPs

In general, transductive models are not convenient for practical use. Each time new test data is seen, the EM algorithm needs to be re-run. A more flexible ("inductive") model can be derived from the following theorem:

Theorem 5.2. *Given $\boldsymbol{\mu}_\mathbf{f}$ and $\mathbf{K}$ sampled from the hyper prior specified in Eq. (10), 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 normal-inverse-Wishart distribu-*

[3] All proofs are given in the appendix.

tion with scale matrix $\boldsymbol{\kappa}^{-1}$*:*

$$p(\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha) = \mathcal{N}(\boldsymbol{\mu}_\alpha|0, \frac{1}{\pi}\mathbf{C}_\alpha)\mathcal{IW}(\mathbf{C}_\alpha|\tau, \boldsymbol{\kappa}^{-1}) \tag{11}$$

This suggests the following equivalent form of Model 2:

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 5.2, define as the generative model:*

1. $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ *are generated once Eq. (11);*
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* $\epsilon \sim \mathcal{N}(0, \sigma^2)$*,* $\mathbf{x}_i \in \mathbf{X}$*.*

The estimates of $\theta = \{\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha, \sigma^2\}$ and $\boldsymbol{\alpha}^l$ can be again learned via the following EM algorithm:[4] The complexity is now independent of dimensionality d of data, but dependent on the training size n, which is $O(kmn^3)$. When the training size is much smaller than dimensionality (the case in our experiment on text categorization), Model 3 is much more efficient then Model 1.

- *E-step*: Estimate the expectation and covariance of $\boldsymbol{\alpha}^l$, $l = 1, \ldots, m$, given the current θ. .

$$\hat{\boldsymbol{\alpha}}^l = \Big(\frac{1}{\sigma^2}\boldsymbol{\kappa}_l^\intercal\boldsymbol{\kappa}_l + \mathbf{C}_\alpha^{-1}\Big)^{-1}\Big(\frac{1}{\sigma^2}\boldsymbol{\kappa}_l^\intercal\mathbf{y}_l + \mathbf{C}_\alpha^{-1}\boldsymbol{\mu}_\alpha\Big) \tag{12}$$

$$\mathbf{C}_{\alpha^l} = \Big(\frac{1}{\sigma^2}\boldsymbol{\kappa}_l^\intercal\boldsymbol{\kappa}_l + \mathbf{C}_\alpha^{-1}\Big)^{-1} \tag{13}$$

  where $\boldsymbol{\kappa}_l \in \mathbb{R}^{n_l \times n}$ is the base kernel $\kappa(\cdot,\cdot)$ evaluated between $\mathbf{X}_l$ and $\mathbf{X}$.

- *M-step*: Optimize $\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha$ and σ.

$$\boldsymbol{\mu}_\alpha = \frac{1}{\pi + m}\sum_l \hat{\boldsymbol{\alpha}}^l \tag{14}$$

$$\mathbf{C}_\alpha = \frac{1}{\tau + m}\Big\{\pi\boldsymbol{\mu}_\alpha\boldsymbol{\mu}_\alpha^\intercal + \tau\boldsymbol{\kappa}^{-1} + \sum_l \mathbf{C}_{\alpha^l} + \sum_l \left[\hat{\boldsymbol{\alpha}}^l - \boldsymbol{\mu}_\alpha\right]\left[\hat{\boldsymbol{\alpha}}^l - \boldsymbol{\mu}_\alpha\right]^\intercal\Big\} \tag{15}$$

$$\sigma^2 = \frac{1}{\sum_l n_l}\sum_l \|\mathbf{y}_l - \boldsymbol{\kappa}_l\hat{\boldsymbol{\alpha}}^l\|^2 + \mathrm{tr}[\boldsymbol{\kappa}_l\mathbf{C}_{\alpha^l}\boldsymbol{\kappa}_l^\intercal] \tag{16}$$

After the EM algorithm we obtain $\theta = \{\boldsymbol{\mu}_\alpha, \mathbf{C}_\alpha, \sigma^2\}$, $\{\hat{\boldsymbol{\alpha}}^l\}$ and the estimated *inductive* functions

$$\hat{f}_l(\mathbf{x}) = \sum_{i=1}^n \hat{\alpha}_i^l \kappa(\mathbf{x}_i, \mathbf{x}) \tag{17}$$

[4]The derivation is essentially the same as that for Model 1, and is thus omitted.

Very importantly, the following theorem justifies the estimated functions in Eq. (17).

Theorem 5.3. *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 holds:*

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

According to Theorem 5.3, a certain modification to the linear model 1 does not change the estimates of functions, and meanwhile the newly derived model becomes *identical* to the inductive multi-task GP Model 3. The message is that, working with a finite model Model 3 will give *exactly the same* estimates of functions f_l achieved in the infinite dimensional case of 1. Furthermore, the theorem also indicates that Model 3 will give the same $\hat{f}_l$ as long as $\cup\mathbf{X}_l \subseteq \mathbf{X}$, which suggests we can just set $\mathbf{X} = \cup\mathbf{X}_l$ in Model 3 to achieve the highest efficiency.

According to Theorem 5.2, the GP kernel matrix on the finite points $\mathbf{X}$ is restored as $\mathbf{K} = \boldsymbol{\kappa}\mathbf{C}_\alpha\boldsymbol{\kappa}$. However, the general kernel function $K(\cdot,\cdot)$ is still unknown, but can be approximated by:

$$K(\mathbf{x}_i, \mathbf{x}_j) \approx \kappa(\mathbf{x}_i, \cdot)^\intercal\mathbf{C}_\alpha\kappa(\mathbf{x}_j, \cdot) \tag{18}$$

where $\kappa(\mathbf{x}, \cdot) = [\kappa(\mathbf{x}, \mathbf{x}_1), \ldots, \kappa(\mathbf{x}, \mathbf{x}_n)]^\intercal$. The learned kernel will be helpful in learning new functions. Since the approximation is finite-dimensional, we use the base kernel to compose a valid positive definite kernel function

$$K(\mathbf{x}_i, \mathbf{x}_j) \approx \frac{\left[m\kappa(\mathbf{x}_i, \cdot)^\intercal\mathbf{C}_\alpha\kappa(\mathbf{x}_j, \cdot) + \tau\kappa(\mathbf{x}_i, \mathbf{x}_j)\right]}{\tau + m}. \tag{19}$$

The composition is rather empirical, but the intuition behind is to weight the learned covariance function and the base kernel by their corresponding equivalent sample sizes.

6. Experiments

6.1. A Toy Problem

To illustrate how the presented models can learn covariance functions, we reproduce the toy problem of (Schwaighofer et al., 2005). The data in Fig. 1(a) are samples from a Gaussian process with neural network covariance function, each of the $M = 20$ scenarios corresponds to one "noisy line" of points. The underlying

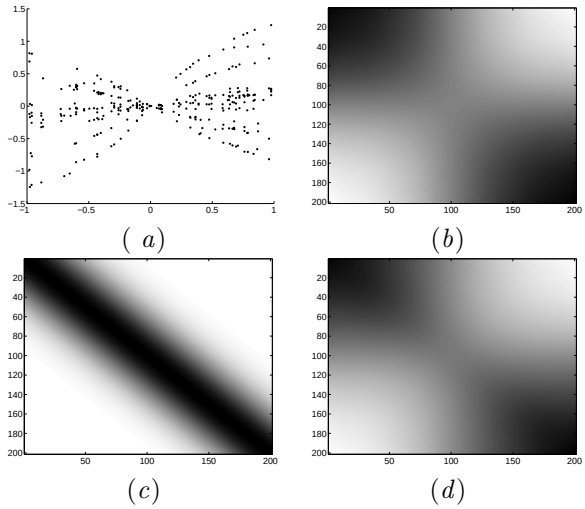


Figure 1. Kernels learned from multiple functions: (a) toy data—multiple functions; (b) true kernel matrix; (c) base kernel matrix; (d) learned kernel matrix.

covariance function is visualized in Fig. 1(b), by evaluating the true covariance on a grid on $[-1, 1]$. Model 3 was used to learn on this data, with the base kernel shown in Fig. 1(c). The "learned covariance" Eq. (18) is visualized in Fig. 1(d), again by plotting the covariance matrix on the $[-1, 1]$ grid.

6.2. Text Categorization

We next consider multi-task learning on a subset of the RCV1-v2 text data set, provided by Reuters and further processed by (Lewis et al., 2004). Since it is common that one document is assigned to multiple topics, this is an ideal data set for multi-task learning, with a binary classifier for each category. As preprocessing, we pick 10000 documents (all topics with more than 50 examples), with a total of 81 categories, and use TFIDF features. On average, each category contains 180 positive documents, and each document belongs to 3.96 topics.

In the first experimental setting, we aim to learn binary classifiers using Model 3 for 50 selected categories. The training example set is fixed with 1000 examples, from which 300 examples for each category are randomly selected to be labeled as $+1$ (in this class) or -1 (not in this class). The remaining examples are left as unlabeled. Each category has its own labeled example set, which is different from each other. The learned classifiers are then used to predict the labels of all the unlabeled examples. In this case the test set contains 9700 examples for each class. However, we distinguish two settings ALL and PARTIALLY LABELED. The first is the evaluation on all the test points, while the second is on those examples with at least one label in some category. We use three common metrics, namely AUC (area under ROC curve), micro-averaged F-value, and macro-averaged F-value. AUC mainly reflects the ranking quality of predictions. The both F-values measure the classification accuracy in the situation of unbalanced classes. In particular, micro-averaged F-value reflects the quality on the classes with more positive examples, while macro-averaged F-value emphasizes on the minor classes. For all the three criteria, larger value indicates better performance. We compare multi-task GP, regularized multi-task learning (Evegniou & Pontil, 2004), linear ridge regression and linear support vector machine (SVM-light, Joachims, 1998). For the multi-task GP, π is set to a large value, to constrain the mean function of the learned GP to zero, and $\tau = 1$. The base kernel is just the linear inner product. For the other algorithms, parameters are set to the values that optimize error rates in cross-validation on the training data. We randomize the setting for 10 times. The averaged results are reported in Tab. 1. Multi-task learning outperforms the other two algorithms with respect to all performance criteria, in particular the performance on the partially labeled data is significantly better. The poor performance of SVMs can be explained in part by the fact that the SVM hinge loss is not well suited for unbalanced data (Zhang & Oles, 2001). Regularized multi-task learning performs similar to SVM. We attribute this to the fact that there is no shared mean function over these categorization classifiers, thus regularized multi-task learning models each function independently.

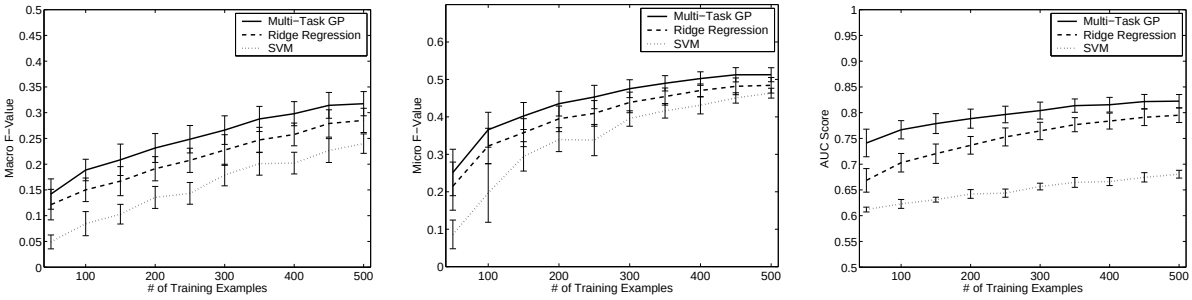
In the second experimental setting, we test the learned kernel Eq. (19) on 31 new categories. The results are shown in Fig. 2, where the method labeled 'Multi-task GP' is in fact kernel ridge regression using the kernel learned by the multi-task GP. We increase the training size from 10 to 500 and randomize 50 times by choosing the learned kernel (recall that there are 10 choices from the first experimental setting) and the training set. Mean performance and error bars are visualized. The results demonstrate that the multi-task GP learns an implicit feature mapping from previously handled tasks, and then generalizes to new, related tasks.

7. Summary and Conclusions

In this paper, we presented a Gaussian process approach to HB learning, by employing the correspondence between parametric linear models and the equivalent Gaussian process model. The final model, presented in Sec. 5, is a both efficient and accurate tool

Table 1. Comparison of four algorithms for 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
Regularized Multi-Task Learning	0.701	0.571	0.232	0.709	0.545	0.216
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



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

Figure 2. Generalization of learned kernels on new categories

for solving multi-task learning problems, as our experiments on multi-label text categorization suggest. We emphasize that, for the models presented here, no form of complex optimization problem needs to be solved, so that real-world multi-task problems can be handled. Also, note that the inductive model 3 is learning a new kernel function that captures information collected from the individual data sets in a compact form, that can be used in further learning tasks.

Acknowledgments

Thanks to Dr. Olivier Chapelle for pointing out his related work on conjoint analysis. Also, thanks to the reviewers for constructive comments.

References

Ando, R. K., & Zhang, T. (2004). A framework for learning predictive structures from multiple tasks and unlabeled data. Technical Report RC23462, IBM T.J. Watson Research Center.

Bakker, B., & Heskes, T. (2003). Task clustering and gating for bayesian multitask learning. *Journal of Machine Learning Research*, *4*, 83–99.

Blei, D. M., Ng, A. Y., & Jordan, M. I. (2003). Latent Dirichlet allocation. *Journal of Machine Learning Research*, *3*, 993–1022.

Caruana, R. (1997). Multitask learning. *Machine Learning*, *28*, 41–75.

Chapelle, O., & Harchaoui, Z. (2005). A machine learning approach to conjoint analysis. *Neural Information Processing Systems 17* (pp. 257–264).

Evegniou, T., & Pontil, M. (2004). Regularized multi–task learning. *Proc. of 17–th SIGKDD Conf. on Knowledge Discovery and Data Mining.*

Gelman, A., Carlin, J., Stern, H., & Rubin, D. (1995). *Bayesian data analysis.* Texts in Statistical Science. Chapman & Hall. First CRC Press reprint 2000.

Gupta, A. K., & Naga, D. K. (1999). *Matrix variate distributions.* No. 104 in Monographs and Surveys in Pure and Applied Mathematics. Chapman & Hall.

Joachims, T. (1998). Text categorization with support vector machines: Learning with many relevant features. *Machine Learning: ECML-98 10th European Conference on Machine Learning* (pp. 137–142). Springer.

Lawrence, N. D., & Platt, J. C. (2004). Learning to learn with the informative vector machine. *Proceedings of ICML04.* Morgan Kaufmann.

Lewis, D. D., Yang, Y., Rose, T. G., & Li, F. (2004). Rvc1: A new benchmark collection for text categorization research. *Journal of Mashine Learning Research*, *5*, 361–397.

Schwaighofer, A., Tresp, V., & Yu, K. (2005). Hierarchical bayesian modelling with gaussian processes.

Advances in Neural Information Processing Systems 17. MIT Press. Accepted for publication.

Williams, C. K. (1998). Prediction with gaussian processes: From linear regression to linear prediction and beyond. In M. I. Jordan (Ed.), *Learning in graphical models*. MIT Press.

Zhang, T., & Oles, F. J. (2001). Text categorization based on regularized linear classification methods. *Information Retrieval*, *4*, 5–31.

A. Derivation of the EM Algorithm for Model 1

The joint distribution of $\mathbf{y}_l$ and $\mathbf{w}_l$ given θ is

$$p(\{\mathbf{y}_l\},\{\mathbf{w}_l\}|\{\mathbf{X}_l\},\theta)=\prod_l \frac{1}{Z_l}\exp\big(-\frac{1}{2}J(\mathbf{w}_l)\big)$$

where Z_l is a normalization term, and the exponential term $J(\mathbf{w}_l)$ equals to

$$\frac{1}{\sigma^2}\sum_{i\sim l}(\mathbf{w}_l^\intercal\mathbf{x}_i-y_i^l)^2+(\mathbf{w}_l-\boldsymbol{\mu}_w)^\intercal\mathbf{C}_w^{-1}(\mathbf{w}_l-\boldsymbol{\mu}_w)$$

Based on the joint distribution and Bayes rule, at the E-step, since the *a posteriori* distribution of latent variables $\mathbf{w}_l$ as a product of m Gaussian posteriori distributions, we compute the sufficient statistics of each Gaussian. The expectation of $\mathbf{w}_l$ is obtained by setting $\frac{\partial J(\mathbf{w}_l)}{\partial \mathbf{w}_l}=0$, which leads to

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

The covariance of $\mathbf{w}_l$ can be derived from computing the inverse of Hessian $\mathbf{C}_{w_l}=\big(\frac{\partial J(\mathbf{w}_l)}{\partial\mathbf{w}_l\partial\mathbf{w}_l^\intercal}\big)^{-1}=\Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal\mathbf{X}_l+\mathbf{C}_w^{-1}\Big)^{-1}$ At the M-step, we optimize $\boldsymbol{\mu}_w,\mathbf{C}_w,\sigma^2$ to maximize the penalized expected log-likelihood of complete data over the a posteriori distribution estimated from the E-step. The negative log-likelihood of complete data is

$$\begin{aligned}&-\ln p(\{\mathbf{y}_l\},\{\mathbf{w}_l\}|\boldsymbol{\mu}_w,\mathbf{C}_w,\sigma)\\&=\frac{1}{2}\sum_l\Big[(n_l+d)\ln 2\pi+n_l\ln(\sigma^2)+\ln|\mathbf{C}_w|\\&+\frac{1}{\sigma^2}\|\mathbf{y}_l-\mathbf{X}_l\mathbf{w}_l\|^2+(\mathbf{w}_l-\boldsymbol{\mu}_w)^\intercal\mathbf{C}_w^{-1}(\mathbf{w}_l-\boldsymbol{\mu}_w)\Big]\end{aligned}$$

The corresponding expectation is

$$\begin{aligned}Q(\theta)=\text{const}&+\frac{m\ln|\mathbf{C}_w|}{2}+\sum_l n_l\ln\sigma\\&+\frac{1}{2}\sum_l\Big\{\mathrm{E}\Big[\mathbf{w}_l^\intercal\Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal\mathbf{X}_l+\mathbf{C}_w^{-1}\Big)\mathbf{w}_l\Big]\\&-2\mathrm{E}\Big[\mathbf{w}_l^\intercal\Big(\frac{1}{\sigma^2}X_l^\intercal\mathbf{y}_l+\mathbf{C}_w^{-1}\boldsymbol{\mu}_w\Big)\Big]\\&+\frac{1}{\sigma^2}\mathbf{y}_l^\intercal\mathbf{y}_l+\boldsymbol{\mu}_w^\intercal\mathbf{C}_w^{-1}\boldsymbol{\mu}_w\Big\}\end{aligned}$$

where const $=\frac{1}{2}\sum_l(n_l+d)\ln 2\pi$, and the two expectations are

$$\begin{aligned}&\mathrm{E}\Big[\mathbf{w}_l^\intercal\Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal\mathbf{X}_l+\mathbf{C}_w^{-1}\Big)\mathbf{w}_l\Big]=\\&\mathrm{Tr}\Big[\Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal\mathbf{X}_l+\mathbf{C}_w^{-1}\Big)\mathbf{C}_{w_l}\Big]+\hat{\mathbf{w}}_l^\intercal\Big(\frac{1}{\sigma^2}\mathbf{X}_l^\intercal\mathbf{X}_l+\mathbf{C}_w^{-1}\Big)\hat{\mathbf{w}}_l\\&\mathrm{E}\Big[\mathbf{w}_l^\intercal\Big(\frac{1}{\sigma^2}X_l^\intercal\mathbf{y}_l+\mathbf{C}_w^{-1}\boldsymbol{\mu}_w\Big)\Big]=\hat{\mathbf{w}}_l\Big(\frac{1}{\sigma^2}X_l^\intercal\mathbf{y}_l+\mathbf{C}_w^{-1}\boldsymbol{\mu}_w\Big)\end{aligned}$$

The updates of θ are then achieved by $\arg\max_\theta Q(\theta)+\ln p(\boldsymbol{\mu}_w,\mathbf{C}_w)$ where $p(\boldsymbol{\mu}_w,\mathbf{C}_w)$ is the hyper prior. The maximization is very straightforward and has to be omitted due to the space limitation here.

B. Proofs

In the proofs we often use the following property of the Wishart distribution (Gupta & Naga, 1999, Theorem 3.3.11):

Lemma 1. *For $A\in\mathbb{R}^{p\times p}$, $\mathbf{A}\sim\mathcal{IW}(\tau,\Lambda)$, and nonsingular $\mathbf{B}\in\mathbb{R}^{p\times q}$, then $\mathbf{B}^\intercal\mathbf{A}\mathbf{B}\sim\mathcal{IW}(\tau,\mathbf{B}^\intercal\Lambda\mathbf{B})$.*

B.1. Theorem 5.1

Proof. This is a simple application of Lemma 1. Since $\mathbf{X}$ is always a linearly independent set, $\boldsymbol{\kappa}=\mathbf{X}\mathbf{X}^\intercal\succ 0$ and $\mathbf{X}$ is nonsingular. Given the normal-inverse-Wishart distribution defined in Eq. (4), the new base covariance matrix $\mathrm{E}(\mathbf{f}^\intercal\mathbf{f})=\mathbf{X}\mathbf{C}_{\mathrm{w}_0}\mathbf{X}^\intercal=\boldsymbol{\kappa}$ and the covariance of $\boldsymbol{\mu}_\mathbf{f}$ clearly becomes $\frac{1}{\pi}\mathbf{K}$, which completes the proof. □

B.2. Theorem 5.2

Proof. (1) Given $\boldsymbol{\kappa}\succ 0$, there are unique $\boldsymbol{\mu}_\alpha=\boldsymbol{\kappa}^{-1}$ and $\mathbf{C}_\alpha=\boldsymbol{\kappa}^{-1}\mathbf{K}\boldsymbol{\kappa}^{-1}$; (2) Again, since $\boldsymbol{\kappa}\succ 0$, $\boldsymbol{\alpha}=\boldsymbol{\kappa}^{-1}\mathbf{f}$, $\mathrm{E}(\boldsymbol{\alpha})=\boldsymbol{\kappa}^{-1}\mathrm{E}(\mathbf{f})=\boldsymbol{\mu}_\alpha$ and $\mathbf{C}_\alpha=\mathrm{E}(\boldsymbol{\alpha}\boldsymbol{\alpha}^\intercal)=\boldsymbol{\kappa}^{-1}\mathrm{E}(\mathbf{f}\mathbf{f}^\intercal)\boldsymbol{\kappa}^{-1}$; (3) The conclusion simply follows from Lemma 1, since $\mathbf{C}_\alpha=\boldsymbol{\kappa}^{-1}\mathbf{K}\boldsymbol{\kappa}^{-1}$, the corresponding base covariance $\mathbf{C}_{\alpha_0}=\mathbf{C}_\alpha=\boldsymbol{\kappa}^{-1}\mathbf{C}_\mathbf{f}\boldsymbol{\kappa}^{-1}=\boldsymbol{\kappa}^{-1}$. □

B.3. Theorem 5.3

Proof. For $\mathbf{w}\in S$, since $\mathbf{X}$ is nonsingular and thus composed by a set of linearly independent columns, there is a unique $\boldsymbol{\alpha}$ satisfying $\mathbf{w}=\mathbf{X}^\intercal\boldsymbol{\alpha}$. Then we have $f(\mathbf{x})=\mathbf{w}^T\mathbf{x}=\sum_{i=1}^n(\boldsymbol{\alpha})_i\kappa(\mathbf{x}_i,\mathbf{x})$ and the function values on $\mathbf{X}$ with the form $\mathbf{f}=\boldsymbol{\kappa}\boldsymbol{\alpha}$. Furthermore, there is $\mathbf{K}=\mathrm{E}(\mathbf{f}\mathbf{f}^\intercal)=\mathbf{X}\mathrm{E}(\mathbf{w}\mathbf{w}^\intercal)\mathbf{X}^\intercal=\boldsymbol{\kappa}\mathrm{E}(\boldsymbol{\alpha}\boldsymbol{\alpha}^\mathrm{T})\boldsymbol{\kappa}=\boldsymbol{\kappa}\mathbf{C}_\alpha\boldsymbol{\kappa}$ which gives $\mathbf{C}_\alpha=\boldsymbol{\kappa}^{-1}\mathbf{K}\boldsymbol{\kappa}^{-1}$. Then it is clear that Theorem 5.2 as well as Model 3 apply here, which finishes the first part. For the second part, we first show that the mean $\boldsymbol{\mu}_w$ lies S. There is a decomposition $\boldsymbol{\mu}_w=\mathbf{P}\boldsymbol{\mu}_w+(\mathbf{I}-\mathbf{P})\mu_w=\boldsymbol{\mu}_\parallel+\boldsymbol{\mu}_\perp$. The part $\boldsymbol{\mu}_\perp$ is orthogonal to labeled examples $\cup\mathbf{X}_l$, thus has no impact on the likelihood but just decreases the probability $p(\boldsymbol{\mu}_w)$. Thus, this term vanishes at the optimum, which means $\boldsymbol{\mu}_w\in S$. Then, any function weight $\mathbf{w}\in\mathcal{N}(\boldsymbol{\mu}_w,\mathbf{C}_w)$ can be decomposed as $\mathbf{w}=\boldsymbol{\mu}_w+\mathbf{v}$, where the offset $\mathbf{v}\sim\mathcal{N}(0,\mathbf{C}_w)$ can be further decomposed as $\mathbf{v}=\mathbf{P}\mathbf{v}+(\mathbf{I}-\mathbf{P})\mathbf{v}=\mathbf{v}_\parallel+\mathbf{v}_\perp$. For the same reason $\mathbf{v}_\perp$ also vanishes at the optimum. Therefore we simply restrict $\boldsymbol{\mu}_\perp$ and $\mathbf{v}_\perp=0$ and still obtain the same estimates $\hat{\mathbf{w}}_l$. □