PSEUDO INPUTS FOR PAIRWISE LEARNING WITH GAUSSIAN PROCESSES

Jens Brehm Nielsen, Bjørn Sand Jensen and Jan Larsen

DTU Informatics
Technical University of Denmark
Asmussens Alle B305, 2800 Kgs. Lyngby, Denmark
{jenb,bjje,jl}@imm.dtu.dk

ABSTRACT

We consider learning and prediction of pairwise comparisons between instances. The problem is motivated from a perceptual view point, where pairwise comparisons serve as an effective and extensively used paradigm. A state-of-the-art method for modeling pairwise data in high dimensional domains is based on a classical pairwise probit likelihood imposed with a Gaussian process prior. While extremely flexible, this non-parametric method struggles with an inconvenient $\mathcal{O}\left(n^3\right)$ scaling in terms of the n input instances which limits the method only to smaller problems. To overcome this, we derive a specific sparse extension of the classical pairwise likelihood using the pseudo-input formulation. The behavior of the proposed extension is demonstrated on a toy example and on two real-world data sets which outlines the potential gain and pitfalls of the approach. Finally, we discuss the relation to other similar approximations that have been applied in standard Gaussian process regression and classification problems such as FI(T)C and PI(T)C.

1. INTRODUCTION

The pairwise learning setting has several application areas such as preference learning and ranking [1], metric learning [2] and general pairwise comparison paradigms. Pairwise comparisons are naturally motivated from a perceptual point of view, where human subjects make a sequence of pairwise (subjective) preference decisions in relation to sound quality, music taste, etc. The main advantage is that pairwise relations are relatively easy for subjects to convey consistently since subjects do not need an internal reference.

The theory underlying pairwise comparisons was first formulated in a principle manner in [3] stating *The Law of Comparative Judgments* building on cognitive and perceptual ideas. The basic idea is that a choice is determined by the difference in the response from a latent stochastic process. The resulting likelihood function in its simplest form—which is also by far the most common one—was first put into the flexible framework of Gaussian processes priors in [4].

Gaussian process based models are flexible and thus desirable for pairwise learning, but struggle with an inconvenient $\mathcal{O}\left(n^3\right)$ scaling in terms of the number of input instances n. This makes their use impractical for large-scale problems. Several suggestions have been proposed to remedy this issue for the standard Gaussian process regression case by using a smaller set of inputs that is either a subset of the original input set [5, 6] or a completely new set of *pseudo inputs* [7, 8, 9]. An unifying view of the latter family of models is given in [10] and extended in [11] leading to the well-known FI(T)C and PI(T)C approximations for standard regression and classification models.

In the standard case the explicit formulation of pseudo inputs can easily and without further considerations be turned into a conditional Gaussian process prior with an easy to invert covariance matrix. However, in the pairwise case the likelihood function depends on two variables. Therefore, we cannot immediately and without consideration use the standard approximations in the covariance as done in [12]. Instead, our quest to derive a sparse approximation for pairwise problems starts from the original pseudo-input formulation presented in [7]. Using this direct approach, our objective is to extend the pairwise likelihood model to allow for explicit sparsity in input space achieved by extending the model by a set of pseudo inputs—or inducing points—of size $l \ll n$. Essentially, the pseudo inputs are used to integrate out the two original variables of the classical pairwise likelihood function. In effect the Gaussian process prior is now placed over the function values of the pseudo inputs often resulting in a considerably lower computational load. Posterior inference relies on a Laplace approximation and the pseudo inputs can be found by evidence optimization for example initialized by k-means.

We give insight and intuition about the behavior and performance of the sparse model compared with the standard model by considering the *Boston housing* data set and a *wine-quality* data set. Examination of the out-of-sample error rates

This work was supported in part by the IST Programme of the European Community, under the PASCAL2 Network of Excellence, IST-2007-216886. This publication only reflects the authors' views.

978-1-4673-1026-0/12/$31.00 ©2012 IEEE

is the basis for discussing the potential and limitations of the sparse model.

2. MODEL & EXTENSIONS

In this section we describe the general setup and frame the pairwise model in a Bayesian non-parametric setting. Each input instance i is described by a feature vector $\mathbf{x} \in \mathbb{R}^d$ and $\mathcal{X} = \{\mathbf{x}_i | i = 1, ..., n\}$. Next, we consider a data set $\mathcal{Y} = \{y_k; u_k, v_k | k = 1, ..., m\}$ of pairwise relations $y \in \{-1, +1\}$ between the u'th and the v'th instance of $\mathcal{X}$, hence $\mathbf{x}_{u_k}, \mathbf{x}_{v_k} \in \mathcal{X}$ [1] . The two opposite choices picking either the u'th or the v'th instance are denoted by $y = -1$ and $y = +1$, respectively.

Given two latent function values $\mathbf{f}_k = [f(\mathbf{x}_{u_k}), f(\mathbf{x}_{v_k})]^\top$, the observations are modeled by a pairwise likelihood function $p\left(y_k | \mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right)$ with parameter(s) $\boldsymbol{\theta}_{\mathcal{L}}$. The function f is an latent function, which in a Thurstonian context [13], models the mean absolute response from the internal cognitive process when the subject is exposed to an input instance. The function parametrization admits that we directly place a zero-mean Gaussian process [14] prior on f allowing for a flexible predictive model for the pairwise responses. Formally, we write $f(\mathbf{x}_i) \sim \mathcal{GP}\left(0, k_{\boldsymbol{\theta}_{\mathcal{GP}}}(\mathbf{x}_i, \cdot)\right)$, where $k(\cdot, \cdot)$ denotes a covariance function, or kernel, with parameter(s) $\boldsymbol{\theta}_{\mathcal{GP}}$, which generally speaking restricts the smoothness of the function. The fundamental consequence of a Gaussian process is that the joint distribution of a finite set of function values $\mathbf{f} = [f(\mathbf{x}_1), f(\mathbf{x}_2), f(\mathbf{x}_3), ..., f(\mathbf{x}_n)]^\top$ has a multivariate Gaussian distribution defined by $p\left(\mathbf{f} | \mathcal{X}, \boldsymbol{\theta}_{\mathcal{GP}}\right) = \mathcal{N}\left(\mathbf{0}, \mathbf{K}_{\mathcal{X}\mathcal{X}}\right)$, where the elements of the covariance matrix are given as $[\mathbf{K}_{\mathcal{X}\mathcal{X}}]_{i,j} = k_{\boldsymbol{\theta}_{\mathcal{GP}}}(\mathbf{x}_i, \mathbf{x}_j)$. Given a standard Bayesian framework and assuming i.i.d. comparisons we now obtain the posterior over the function values

$$p\left(\mathbf{f} | \mathcal{X}, \mathcal{Y}, \boldsymbol{\theta}\right) \propto p\left(\mathbf{f} | \mathcal{X}, \boldsymbol{\theta}_{\mathcal{GP}}\right) \prod\nolimits_{k=1}^{m} p\left(y_k | \mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right)$$

with $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\mathcal{L}}, \boldsymbol{\theta}_{\mathcal{GP}}\}$. The main computational issue in the Gaussian process framework is to calculate/approximate the posterior posing a $\mathcal{O}\left(n^3\right)$ scaling challenge due to the inversion of the kernel matrix.

2.1. Standard Pairwise Likelihood Function

The pairwise likelihood function described in a general pairwise context by [13] and used with Gaussian processes by e.g. [4] and [15] is given by

$$p\left(y_k | \mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right) = \Phi\left(y_k \frac{f\left(\mathbf{x}_{u_k}\right) - f\left(\mathbf{x}_{v_k}\right)}{\sqrt{2}\sigma}\right), \qquad (1)$$

where $\Phi(\cdot)$ defines a cumulative Gaussian (with zero mean and unity variance) and $\boldsymbol{\theta}_{\mathcal{L}} = \{\sigma\}$. The use of a Gaussian process prior in connection with this likelihood function was first proposed in [4].

2.2. Sparse Pairwise Likelihood Function

To obtain sparsity in input space, we generally follow the ideas in [7]. Hence, given a set of pseudo inputs $\bar{\mathbf{X}}$, their functional values $\bar{\mathbf{f}}$ must originate from the same Gaussian process that was used for $\mathbf{f}$. Therefore, we can directly place a Gaussian process prior over $\bar{\mathbf{f}}$, i.e., $p\left(\bar{\mathbf{f}} | \bar{\mathbf{X}}\right) = \mathcal{N}\left(\bar{\mathbf{f}} | \mathbf{0}, \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}\right)$, where the matrix $\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}$ is the covariance matrix of the l pseudo inputs collected in the matrix $\bar{\mathbf{X}} = [\bar{\mathbf{x}}_1, ..., \bar{\mathbf{x}}_l]$.

The overall idea of the pseudo-input formalism is now to refine the likelihood function from Eq. (1) such that the real $\mathbf{f}$ values that enter directly in the original, non-sparse likelihood function (through $\mathbf{f}_k$), exist only in the form of predictions from the pseudo inputs $\bar{\mathbf{f}}(\bar{\mathbf{X}})$. Given the listed assumptions, we formally have that $\mathbf{f}$ and $\bar{\mathbf{f}}$ are jointly Gaussian, hence

$$\begin{bmatrix} \mathbf{f}_k \\ \bar{\mathbf{f}} \end{bmatrix} = \mathcal{N}\left(\begin{bmatrix} \mathbf{0} \\ \mathbf{0} \end{bmatrix}, \begin{bmatrix} \mathbf{K}_{\mathbf{x}_k\mathbf{x}_k} & \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k}{}^\top \\ \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k} & \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}} \end{bmatrix}\right), \qquad (2)$$

where we define the following matrices and vectors

$$\begin{aligned} \mathbf{K}_{\mathbf{x}_k\mathbf{x}_k} &= \begin{bmatrix} k(\mathbf{x}_{u_k}, \mathbf{x}_{u_k}) & k(\mathbf{x}_{u_k}, \mathbf{x}_{v_k}) \\ k(\mathbf{x}_{v_k}, \mathbf{x}_{u_k}) & k(\mathbf{x}_{v_k}, \mathbf{x}_{v_k}) \end{bmatrix} & (3) \\ \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k} &= [\mathbf{k}_{u_k}, \mathbf{k}_{v_k}] & (4) \end{aligned}$$

with $[\mathbf{k}_{u_k}]_i = k(\bar{\mathbf{x}}_i, \mathbf{x}_{u_k})$ and $[\mathbf{k}_{v_k}]_i = k(\bar{\mathbf{x}}_i, \mathbf{x}_{v_k})$. From Eq. (2) it is trivial to find the conditional distribution of $\mathbf{f}_k$ given $\bar{\mathbf{f}}$, hence the sparse likelihood function can be derived in terms of $\bar{\mathbf{f}}$ by integrating over $\mathbf{f}_k$, thus

$$\begin{aligned} & p\left(y_k | \mathbf{x}_{u_k}, \mathbf{x}_{v_k}, \bar{\mathbf{X}}, \bar{\mathbf{f}}, \boldsymbol{\theta}\right) \\ &= \int p\left(y_k | \mathbf{f}_k, \boldsymbol{\theta}_{\mathcal{L}}\right) p\left(\mathbf{f}_k | \bar{\mathbf{f}}, \bar{\mathbf{X}}\right) d\mathbf{f}_k \\ &= \int \Phi\left(y_k \frac{f\left(\mathbf{x}_{u_k}\right) - f\left(\mathbf{x}_{v_k}\right)}{\sqrt{2}\sigma}\right) \mathcal{N}\left(\mathbf{f}_k | \boldsymbol{\mu}_k, \boldsymbol{\Sigma}_k\right) d\mathbf{f}_k \\ &= \Phi\left(y_k \frac{\mu_{u_k} - \mu_{v_k}}{\sigma_k^*}\right) \end{aligned}$$

where $\boldsymbol{\mu}_k = [\mu_{u_k}, \mu_{v_k}]^\top$, $\mu_{u_k} = \mathbf{k}_{u_k}^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{f}}$, $\mu_{v_k} = \mathbf{k}_{v_k}^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{f}}$ and

$$\boldsymbol{\Sigma}_k = \begin{bmatrix} \sigma_{u_k u_k} & \sigma_{u_k v_k} \\ \sigma_{v_k u_k} & \sigma_{v_k v_k} \end{bmatrix} = \mathbf{K}_{\mathbf{x}_k\mathbf{x}_k} - \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k}^\top \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \mathbf{K}_{\bar{\mathbf{X}}\mathbf{x}_k}$$

Furthermore, $(\sigma_k^*)^2 = 2\sigma^2 + \sigma_{u_k u_k} + \sigma_{v_k v_k} - \sigma_{u_k v_k} - \sigma_{v_k u_k}$, which all together results in the pseudo-input likelihood

$$p\left(y_k | \mathbf{x}_{u_k}, \mathbf{x}_{v_k}, \bar{\mathbf{X}}, \bar{\mathbf{f}}, \boldsymbol{\theta}\right) = \Phi\left(z_k\right), \qquad (5)$$

with $z_k = y_k \left(\mathbf{k}_{u_k}^T - \mathbf{k}_{v_k}^T\right) \mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} \bar{\mathbf{f}} / \sigma_k^*$.

[1] We will without loss of generality assume that the set $\mathcal{Y}$ involves all n inputs instances in $\mathcal{X}$.

2.3. Inference & Predictions

The likelihood functions described in Section 2.1 and 2.2 lead to intractable posteriors and call for approximation techniques or sampling methods. Our goal in this initial study is to examine the sparse model and its properties—not to provide the optimal approximation—hence, we only explore inference based on the Laplace approximation.

2.3.1. *Posterior Approximation*

Inference using the Laplace approximation has also been applied in [16] for the standard model. The general solution to the approximation problem can be found by maximizing the unnormalized log-posterior $\psi\left(\bar{\mathbf{f}}|\mathcal{Y},\mathcal{X},\bar{\mathbf{X}},\boldsymbol{\theta}\right) = \log p\left(\mathcal{Y}|\bar{\mathbf{f}},\mathcal{X},\bar{\mathbf{X}},\boldsymbol{\theta}\right) - \frac{1}{2}\bar{\mathbf{f}}^T\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}\bar{\mathbf{f}} - \frac{1}{2}\log|\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}| - \frac{l}{2}\log 2\pi$ with regards to $\bar{\mathbf{f}}$. For the maximization we use a damped Newton method in which the damped step (with adaptive damping factor λ) can be calculated without inversion of the Hessian

$$\bar{\mathbf{f}}^{new} = \left(\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1} + \mathbf{W} - \lambda\mathbf{I}\right)^{-1} \\ \left[(\mathbf{W}-\lambda\mathbf{I})\,\bar{\mathbf{f}} + \nabla\log p(\mathcal{Y}|\bar{\mathbf{f}},\mathcal{X},\bar{\mathbf{X}},\boldsymbol{\theta})\right]. \quad (6)$$

Using the notation $\nabla\nabla_{i,j} = \frac{\partial^2}{\partial f(x_i)\partial f(x_j)}$ we apply the definition $\mathbf{W}_{i,j} = -\sum_k \nabla\nabla_{i,j}\log p(y_k|\mathbf{x}_{u_k},\mathbf{x}_{v_k},\bar{\mathbf{X}},\bar{\mathbf{f}},\boldsymbol{\theta})$. When converged, the resulting approximation can be shown to be $p\left(\bar{\mathbf{f}}|\mathcal{Y},\mathcal{X},\bar{\mathbf{X}},\boldsymbol{\theta}\right) \approx \mathcal{N}\left(\bar{\mathbf{f}}|\hat{\mathbf{f}},\left(\mathbf{W}+\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}\right)^{-1}\right)$. The damped Newton step requires the Jacobian and Hessian of the new pseudo-input log-likelihood from Eq. (5), which require the following two derivatives

$$\frac{\partial}{\partial\bar{\mathbf{f}}}p\left(y_k|\ldots\right) = y_k\frac{\mathcal{N}\left(z_k\right)}{\sigma_k^*\Phi\left(z_k\right)}\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}\left(\mathbf{k}_{u_k}-\mathbf{k}_{v_k}\right) \quad (7)$$

$$\frac{\partial^2}{\partial\bar{\mathbf{f}}\bar{\mathbf{f}}^\top}p\left(y_k|\ldots\right) = -y_k^2\frac{\mathcal{N}\left(z_k\right)}{(\sigma_k^*)^2\Phi\left(z_k\right)}\left[z_k + \frac{\mathcal{N}\left(z_k\right)}{\Phi\left(z_k\right)}\right] \\ \cdot\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}\left(\mathbf{k}_{u_k}-\mathbf{k}_{v_k}\right)\left(\mathbf{k}_{u_k}-\mathbf{k}_{v_k}\right)^\top\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}. \quad (8)$$

2.3.2. *Evidence / Hyperparameter Optimization*

So far we have simply considered the hyperparameters $\boldsymbol{\theta} = \{\boldsymbol{\theta}_{\mathcal{L}},\boldsymbol{\theta}_{\mathcal{GP}}\}$ and pseudo inputs $\bar{\mathbf{X}}$ as fixed parameters, but their values have a crucial influence on the model performance. Here, we resort to point estimates and find (possible locally) optimal values by iterating between the Laplace approximation with fixed hyperparameters, i.e., finding $p\left(\bar{\mathbf{f}}|\mathcal{Y},\mathcal{X},\bar{\mathbf{X}},\boldsymbol{\theta}\right)$, followed by an evidence maximization step in which $(\boldsymbol{\theta},\bar{\mathbf{X}}) = \arg\max_{(\boldsymbol{\theta},\bar{\mathbf{X}})}p\left(\mathcal{Y}|\boldsymbol{\theta},\bar{\mathbf{X}}\right)$. The log-evidence $\log p(\mathcal{Y}|\boldsymbol{\theta},\bar{\mathbf{X}})$ has to be approximated in our case, which in terms of the existing Laplace approximation yields

$$\log p\left(\mathcal{Y}|\boldsymbol{\theta},\bar{\mathbf{X}}\right) \approx \log q\left(\mathcal{Y}|\bar{\mathbf{X}},\boldsymbol{\theta}\right) = \log p(\mathcal{Y}|\hat{\mathbf{f}},\bar{\mathbf{X}},\mathcal{X},\boldsymbol{\theta}) \\ -\frac{1}{2}\hat{\mathbf{f}}^T\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}^{-1}\hat{\mathbf{f}} - \frac{1}{2}\log|\mathbf{I}+\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}\mathbf{W}|. \quad (9)$$

We further allow for fixed hyperpriors on the individual hyperparameters serving as regularization, which results in a procedure referenced to as MAP-II which provides more robust estimation. Consequently, the MAP-II is given by $\log q_{\text{MAP-II}}\left(\mathcal{Y}|\bar{\mathbf{X}},\boldsymbol{\theta}\right) = \log q\left(\mathcal{Y}|\bar{\mathbf{X}},\boldsymbol{\theta}\right) + \log p\left(\boldsymbol{\theta},\bar{\mathbf{X}}|\xi\right)$, where ξ is a set of fixed parameters in the hyperprior.

The optimization requires the derivatives of the evidence approximation. These turn out to be rather tedious and involved, and we refer to the appendix for details. The pseudo-input model poses a number of difficulties since $\bar{\mathbf{X}}$ are also to be considered hyperparameters. Typically, this will—as noted by [7] and [17]—lead to a large number of local maxima providing potentially suboptimal solutions. It is not our aim to resolve nor document this issue, and we will take a pragmatic view and simply accept evidence optimization methods as is. Like [17] we recommend starting out with a fixed set of pseudo inputs initialized by a standard unsupervised clustering, such as k-means with restarts, followed by evidence optimization.

2.3.3. *Predictions*

The main task is to infer the latent function values $\bar{\mathbf{f}}$ with the end objective to make predictions of the observable variable y for a pair of test inputs $\mathbf{x}_r \in \mathcal{X}_t$ and $\mathbf{x}_s \in \mathcal{X}_t$ denoted $\mathbf{x}_t = [\mathbf{x}_r,\mathbf{x}_s]^T$. We consider the joint distribution between $\bar{\mathbf{f}} \sim p\left(\bar{\mathbf{f}}|\mathcal{Y},\boldsymbol{\theta}\right)$ and the test variables $\mathbf{f}_t = \left[f\left(\mathbf{x}_r\right), f\left(\mathbf{x}_s\right)\right]^T$. With the posterior of $\bar{\mathbf{f}}$ approximated with the Gaussian from the Laplace approximation, the predictive distribution $p\left(\mathbf{f}_t|\mathcal{Y},\boldsymbol{\theta}\right)$ will also be Gaussian given by $\mathcal{N}\left(\mathbf{f}_t|\mu^*,\mathbf{K}^*\right)$ with $\mu^* = [\mu_r^*,\mu_s^*]^T = \mathbf{k}_t\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}{}^{-1}\bar{\mathbf{f}}$ and

$$\mathbf{K}^* = \left[\begin{array}{cc}\sigma_{rr}^* & \sigma_{rs}^* \\ \sigma_{sr}^* & \sigma_{ss}^*\end{array}\right] = \mathbf{K}_t - \mathbf{k}_t^T\left(\mathbf{I}+\mathbf{W}\mathbf{K}_{\bar{\mathbf{X}}\bar{\mathbf{X}}}\right)\mathbf{k}_t,$$

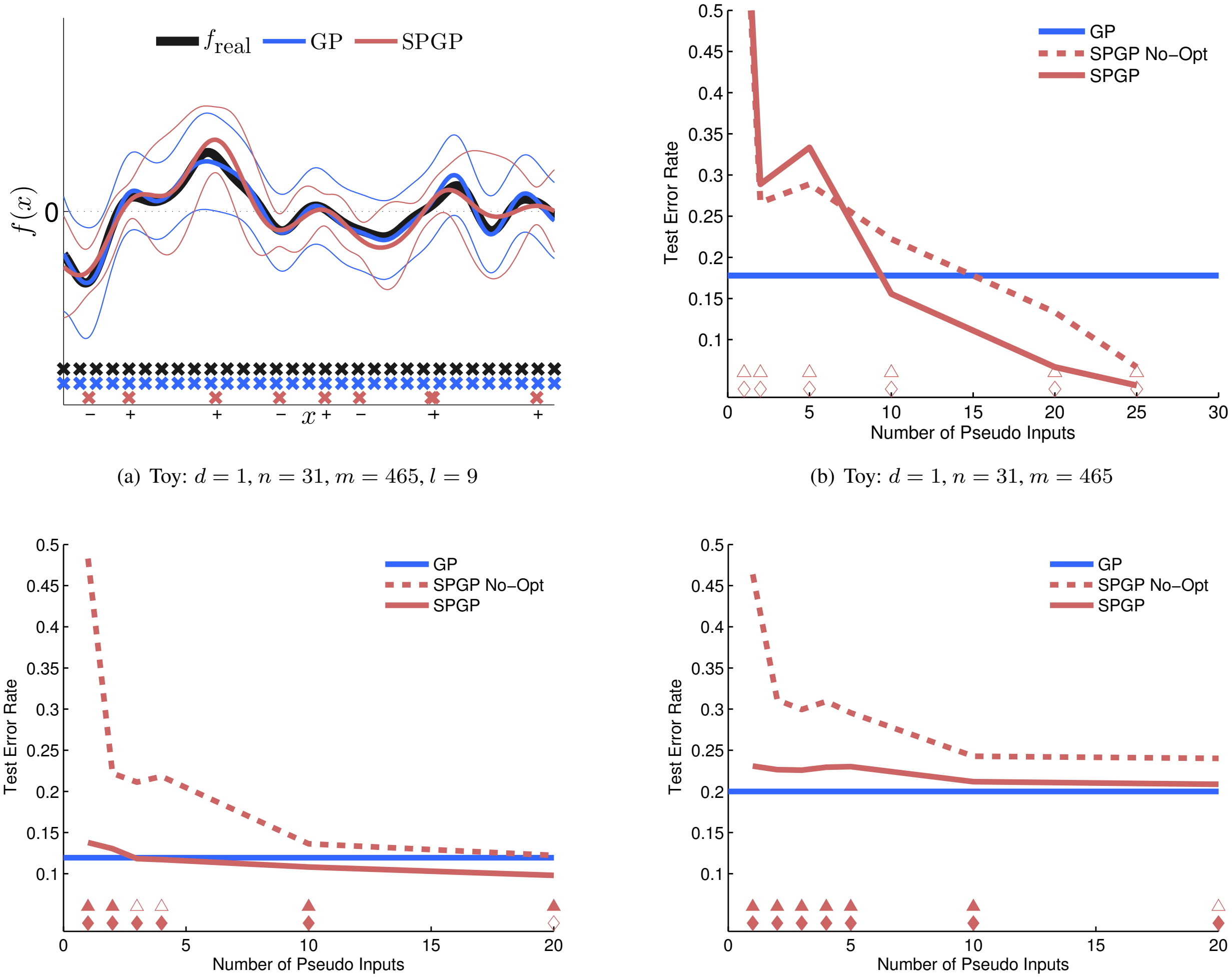
where $\mathbf{k}_t$ is the kernel between the test points and the pseudo inputs. With $p\left(\mathbf{f}_t|\mathcal{Y},\boldsymbol{\theta}\right)$, the prediction distribution of the observed variable is given as $p\left(y_t|\mathcal{Y},\boldsymbol{\theta}\right) = \int p\left(y_t|\mathbf{f}_t,\boldsymbol{\theta}_{\mathcal{L}}\right)p\left(\mathbf{f}_t|\mathcal{Y},\boldsymbol{\theta}\right)d\mathbf{f}_t$. The integral can be calculated in closed form as $P\left(\mathbf{x}_r \succ \mathbf{x}_s|\mathcal{Y},\boldsymbol{\theta}\right) = \Phi\left(\left(\mu_r^*-\mu_s^*\right)/\sigma^*\right)$ with $\left(\sigma^*\right)^2 = 2\sigma^2 + \sigma_{rr}^* + \sigma_{ss}^* - \sigma_{rs}^* - \sigma_{sr}^*$.

3. SIMULATIONS & EXPERIMENTAL RESULTS

In this section we demonstrate the performance of the pseudo-input method on a toy example and provide predictive performance on two real-world data sets: *Boston housing* and *wine quality*. The main objective is not to achieve the overall best performance, but to compare the standard (GP) and the sparse (SPGP) formulations.

3.1. Toy Example

To illustrate the basics of the SPGP model, we draw a deterministic function f_{real} (see Fig. 1(a)) from a zero-mean Gaus-



(a) Toy: $d = 1, n = 31, m = 465, l = 9$

(b) Toy: $d = 1, n = 31, m = 465$

(c) Boston Housing: $d = 10, n = 506, m = 127765$

(d) Wine Quality: $d = 11, n = 600, m = 179700$

Fig. 1. In general, blue graphs indicate the full model (GP) and red indicate the sparse model (SPGP). In **Fig. (a)** thick graphs indicate means and thin graphs indicate one standard deviation. The black graph indicates the real (deterministic) function used to generate the full pairwise data set between the instances marked with black crosses in the bottom. The two other colors sketch the predictive distribution of the GP and SPGP models using the (pseudo) inputs at the locations marked with the corresponding color in the bottom. **Fig. (b)-(d)** display the performance of the sparse model (SPGP) evaluated on the toy example and on the two real-world data sets as a function of the number of pseudo inputs for the sparse model (red). The performance of the standard model is included as a baseline. The **solid** and **dashed** red graphs show the average test error rate for the optimized and non-optimized SPGP model, respectively. The two rows of markers indicate whether the optimized (triangle) and non-optimized (diamond) SPGP models are significant different from the GP model using the McNemar test. The markers are solid if the null hypothesis that they are equal can be rejected at the 5% significance level.

sian process with a squared exponential covariance function. This function is then used to generate a pairwise data set consisting of all possible pairwise comparisons using the function values at equidistantly distributed locations marked with black crosses in Fig. 1(a). To model this data, we consider the two models: The GP model (Sec. 2.1) and the SPGP model with optimized pseudo inputs (Sec. 2.2). The $l = 9$ pseudo inputs are initialized equidistantly in the input interval, the length scale of the covariance function $\boldsymbol{\theta}_{\mathcal{GP}} = \{\sigma_\ell\}$ and the likelihood parameter $\boldsymbol{\theta}_{\mathcal{L}} = \{\sigma\}$ are learned by evidence optimization whereas $\sigma_f = 1$ of the covariance function is fixed. The results are presented in Fig. 1(a).

We notice that the SPGP model is capable of modeling the mean and thereby the actual pairwise relationships, whereas the predictive variance differs significantly from the GP variance. This is a characteristic and expected artifact also seen in connection with the pseudo-input models for standard classification and regression.

3.2. Real World Examples

We compare the performance of the SPGP model to the GP model on two different real-world data sets.

The first data set is the well-known *Boston housing*[2] where we have constructed a full pairwise version by using all $m = 127765$ pairwise combinations of the $n = 506$ inputs base on the house price. For each input we use all available features except RAD, CHAS and NOX, thus $d = 10$.

The second data set is a subset of the *wine quality*[3] which is based on user ratings of wines. The subset is based on $n = 600$ instances of wines described by $d = 11$ features. We construct the set of unique pairwise comparisons from the ratings resulting in $m = 179700$ comparisons.

We use a squared exponential covariance function for both data sets which (based on initial experimentation) is initialized with $\sigma_f = 1$ and $\sigma_\ell = 1$. The covariance parameter σ_f is fixed, whereas the likelihood parameter initialized as $\boldsymbol{\theta}_{\mathcal{L}} = \{\sigma = 1\}$ and $\boldsymbol{\theta}_{\mathcal{GP}} = \{\sigma_\ell\}$ are learned by MAP-II optimization using a uniform hyperprior and a half-student-t hyperprior with scale 6 and 4 degrees of freedom, respectively. Pseudo inputs are initialized with k-means (selecting the solution with minimum total squared distance out of five random initializations). We compare two SPGP models: one where the pseudo inputs are kept fixed following the k-means initialization (this model is identified with the No-Opt tag) and one where they are further fitted using MAP-II with a uniform hyperprior. With both data sets we use 20-fold cross validation on instances, such that a minimum of two instances are held out for testing and a randomly selected quarter of all remaining pairwise comparisons between training instances are used for training. Consequently, predictions are only performed on comparisons between instances that do not appear in the training data and the setting is thus a true predictive ranking scenario. In Fig. 1(c)-(d) we report the average error rate on the test set as a function of the number of pseudo inputs for the two SPGP models. The GP model is included as a baseline.

4. DISCUSSION

In the toy example (Fig. 1(a)) we see that the mean is well modeled by both the GP model and the SPGP model with $l = 9$ pseudo inputs, suggesting that the SPGP model performs nearly as good as the GP model. The main difference between the two models seems to be the predictive variance which differs significantly, yet this is an expected property of the sparse model. A way to improve the estimation of the predictive variance is by allowing the input instances and pseudo inputs to have different length scales [8][17].

Focusing on the predictive mean performance of the optimized SPGP model on the two real-world data sets (Fig. 1(c)-(d)), we see that a SPGP model with few pseudo inputs (as low as 1-5) performs only slightly worse than or equal to the GP model. This indicates that the two real-world problems do not constitute very complex pairwise problems. The performance is, however, highly dependent on the optimization of the locations of the pseudo inputs, seen since the non-optimized SPGP model requires more pseudo inputs due to the fixed locations. This illustrates the importance and power of the optimization.

By further adding pseudo inputs we can obtain better performance than the GP model. We believe that two effects come into play. The first effect is that the constraints induced in the SPGP model provide better regularization compared to the full Gaussian process prior meaning that it generalizes better. The second effect stems from the fact that the arbitrary placement of the pseudo inputs provides added flexibility, which effectively renders it more adequate for capturing the important regions of the underlying function when these locations are optimized appropriately. We speculate that the observed behavior is a combination of the two effects of course dependent on the application.

A further aspect to be investigated is the capability of the SPGP model to capture and approximate higher order moments of the predictive distribution. In line with previous work on the topic and with the variances observed in the toy example, we have observed fluctuating behavior of the predictive likelihoods as a function of l for the SPGP models in the two real-world examples. Whether the behavior is due to the pairwise setting, specific application or a general property of the pseudo-input formulation is an open question.

In the current sparse formulation the original function values are dependent in pairs given the exact comparisons, whereas in FI(T)C all the original function values are independent given the pseudo inputs. We plan to investigate if this difference have any practical importance and to compare the current approximation to other traditional approaches—in particular the PI(T)C approximation.

5. CONCLUSION

In this paper we have derived a sparse version of the pairwise likelihood model using the pseudo-input formulation. We applied the Laplace approximation for both posterior and evidence approximation. We observe competitive predictive performance with the sparse model using only few pseudo inputs on a toy example and on two real-world data sets. A noticeable observation is the fact that we by adding more pseudo inputs are able to obtain better performance than the full GP model in the studied applications.

[2] `archive.ics.uci.edu/ml/datasets/Housing`

[3] `archive.ics.uci.edu/ml/datasets/Wine+Quality`

6. REFERENCES

[1] J. Fürnkranz and E. Hüllermeier, *Preference Learning*, Springer, 1st edition, 2011.

[2] B. McFee and G. Lanckriet, "Metric Learning to Rank," *ICML 2010 - Proceedings, 27th International Conference on Machine Learning*, pp. 775–782, 2010.

[3] L. L. Thurstone, "A Law of Comparative Judgement," *Psychological Review*, vol. 34, 1927.

[4] W. Chu and Z. Ghahramani, "Preference Learning with Gaussian Processes," *Proceedings of the 22nd International Conference on Machine Learning (ICML)*, pp. 137–144, 2005.

[5] N. Lawrence, M. Seeger, and R. Herbrich, "Fast Sparse Gaussian Process Methods: The Informative Vector Machine," in *Neural Information Processing Systems (NIPS)*, 2002, p. 8.

[6] L. Csato, *Gaussian Processes - Iterative Sparse Approximations*, Ph.D. thesis, Aston University, 2002.

[7] E. Snelson and Z. Ghahramani, "Sparse Gaussian Processes using Pseudo-Inputs," *Advances in neural information processing*, 2006.

[8] C. Walder, K. I. Kim, and B. Schölkopf, "Sparse Multiscale Gaussian Process Regression," in *Proceedings of the 25th international conference on Machine Learning*, 2008, pp. 1112–1119.

[9] M. Lazaro-Gredilla and A. Figueiras-Vidal, "Inter-Domain Gaussian Processes for Sparse Inference using Inducing Features," in *Advances in Neural Information Processing Systems 22*, pp. 1087–1095. 2009.

[10] J. Quiñonero-Candela and C.E. Rasmussen, "A Unifying View of Sparse Approximate Gaussian Process Regression," *The Journal of Machine Learning Research*, vol. 6, pp. 1939–1959, 2005.

[11] E. Snelson and Z. Ghahramani, "Local and Global Sparse Gaussian Process Approximations," in *Eleventh International Conference on Artificial Intelligence and Statistics, AISTATS*, 2007, pp. 524–531.

[12] J. Guiver and E. Snelson, "Learning to Rank with Softrank and Gaussian Processes," *Annual ACM Conference on Research and Development in Information Retrieval*, pp. 259–266, 2008.

[13] R. D. Bock and J. V. Jones, "The Measurement and Prediction of Judgment and Choice," 1968.

[14] C. E. Rasmussen and C. K. I. Williams, *Gaussian Processes for Machine Learning*, MIT Press, 2006.

[15] E. Bonilla, S. Guo, and S. Sanner, "Gaussian Process Preference Elicitation," in *Advances in Neural Information Processing Systems 23*.

[16] W. Chu and Z. Ghahramani, "Extensions of Gaussian Processes for Ranking: Semi-Supervised and Active Learning," in *Workshop Learning to Rank at Advances in Neural Information Processing Systems 18*, 2005.

[17] Y. Qi, A. Abdel-Gawad, and T. Minka, "Sparse-Posterior Gaussian Processes for General Likelihoods," in *Proceedings of the Twenty-Sixth Conference Annual Conference on Uncertainty in Artificial Intelligence (UAI-10)*, 2010.

7. APPENDIX - EVIDENCE DERIVATIVES

The derivatives of Eq. (9) are slightly different compared to the standard classification case [14, Sec 5.5.1] due to the pseudo-input model because the covariance parameters enter into the likelihood, and the fact that the covariance function also depends on $\bar{\mathbf{X}}$. We outline the derivations by noting that the Eq. (9) depends both explicitly and implicitly (due to the solution of $\hat{\mathbf{f}}$) on the parameters $\boldsymbol{\theta}$. We do not differentiate between likelihood and covariance parameters and $\bar{\mathbf{X}}$. Here, we simply denote a parameter by θ_j. We can split the derivatives into an explicit and implicit part

$$\frac{\partial \log q\left(\mathcal{Y}|...\right)}{\partial \theta_i} = \left.\frac{\partial \log q\left(\mathcal{Y}|...\right)}{\partial \theta}\right|_{\text{explicit}} + \sum_j \frac{\partial \log q\left(\mathcal{Y}|...\right)}{\partial f_j}\frac{\partial f_j}{\partial \theta_i}.$$

Referring to the **explicit** term we obtain the following terms

$$\frac{\partial}{\partial \boldsymbol{\theta}_i}\log p\left(\mathcal{Y}|\hat{\mathbf{f}},\boldsymbol{\theta}\right)$$

$$\frac{\partial}{\partial \boldsymbol{\theta}_i}\hat{\mathbf{f}}^{\top}{\mathbf{K}_{\boldsymbol{\theta}}}^{-1}\hat{\mathbf{f}} = -\hat{\mathbf{f}}^{\top}\left({\mathbf{K}_{\boldsymbol{\theta}}}^{-1}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_i}{\mathbf{K}_{\boldsymbol{\theta}}}^{-1}\right)\hat{\mathbf{f}}$$

$$\frac{\partial}{\partial \boldsymbol{\theta}_i}\log\left|\mathbf{I}+\mathbf{W}_{\boldsymbol{\theta}}\mathbf{K}_{\boldsymbol{\theta}}\right| = Tr\left[\left(\mathbf{I}+\mathbf{K}_{\boldsymbol{\theta}}\mathbf{W}_{\boldsymbol{\theta}}\right)^{-1}\cdot \left(\frac{\partial \mathbf{W}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_i}\mathbf{K}_{\boldsymbol{\theta}} + \mathbf{W}_{\boldsymbol{\theta}}\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_i}\right)\right]$$

Referring to the **implicit** term we have (without any assumptions regarding the type of parameter)

$$\frac{\partial \log q\left(\mathcal{Y}|\bar{\mathbf{X}},\mathcal{X},\boldsymbol{\theta}\right)}{\partial f_j} = -\frac{1}{2}Tr\left[\left(\mathbf{I}+\mathbf{K}_{\boldsymbol{\theta}}\mathbf{W}_{\boldsymbol{\theta}}\right)^{-1}\left(\mathbf{K}_{\boldsymbol{\theta}}\frac{\partial \mathbf{W}_{\boldsymbol{\theta}}}{\partial f_j}\right)\right]$$

$\frac{\partial f_j}{\partial \boldsymbol{\theta}_i}$ is found by exploiting that $\hat{\mathbf{f}} = \mathbf{K}_{\boldsymbol{\theta}}\nabla \log p\left(\mathcal{Y}|\hat{\mathbf{f}},\boldsymbol{\theta}\right)$ at the current solution leading to the following result

$$\frac{\partial f_j}{\partial \boldsymbol{\theta}_i} = \left(\mathbf{I}+\mathbf{K}_{\boldsymbol{\theta}}\mathbf{W}_{\boldsymbol{\theta}}\right)^{-1}\left(\frac{\partial \mathbf{K}_{\boldsymbol{\theta}}}{\partial \boldsymbol{\theta}_i}\right)\frac{\partial \log p\left(y|\hat{\mathbf{f}},\boldsymbol{\theta}\right)}{\partial \mathbf{f}} + \left(\mathbf{I}+\mathbf{K}_{\boldsymbol{\theta}}\mathbf{W}_{\boldsymbol{\theta}}\right)^{-1}\mathbf{K}_{\boldsymbol{\theta}}\frac{\partial}{\partial \boldsymbol{\theta}_i}\left(\frac{\partial \log p\left(y|\hat{\mathbf{f}},\boldsymbol{\theta}\right)}{\partial \mathbf{f}}\right)$$

We may exploit that the inverse of the common factor $(\mathbf{I}+\mathbf{K}_{\boldsymbol{\theta}}\mathbf{W}_{\boldsymbol{\theta}})$ can be computed using the Cholesky decomposition which enters robustly into the individual expressions for added numerical stability. The expression above is a general result and valid for both likelihood parameters, covariance parameters and pseudo inputs. In addition, the derivatives of the likelihood, Jacobian, Hessian and covariance function are required. One should be aware that some of the derivatives are zero depending on the actual parameter type (e.g. $\partial \mathbf{K}_{\boldsymbol{\theta}}/\partial \boldsymbol{\theta}_{\mathcal{L}}$). The gradients are based on the current Laplace approximation. Even though we take into account implicit dependencies, there is in general no guarantee for strictly monotonic behavior, thus a robust optimization method is required. In practice we have found the BFGS implementation in the `immoptibox`[4] robust.

[4] `www2.imm.dtu.dk/%7Ehbn/immoptibox/`