Journal of Machine Learning Research 6 (2005) 1-48 Submitted 11/04; Published 06/05

Gaussian Processes for Ordinal Regression

Wei Chu CHUWEI@GATSBY.UCL.AC.UK
Zoubin Ghahramani ZOUBIN@GATSBY.UCL.AC.UK
Gatsby Computational Neuroscience Unit
University College London
London, WC1N 3AR, UK

Editor: Christopher K. I. Williams

Abstract

We present a probabilistic kernel approach to ordinal regression based on Gaussian processes. A threshold model that generalizes the *probit* function is used as the likelihood function for ordinal variables. Two inference techniques, based on the Laplace approximation and the expectation propagation algorithm respectively, are derived for hyperparameter learning and model selection. We compare these two Gaussian process approaches with a previous ordinal regression method based on support vector machines on some benchmark and real-world data sets, including applications of ordinal regression to collaborative filtering and gene expression analysis. Experimental results on these data sets verify the usefulness of our approach.

Keywords: Gaussian Processes, Ordinal Regression, Approximate Bayesian Inference, Collaborative Filtering, Gene Expression Analysis, Feature Selection.

1. Introduction

Practical applications of supervised learning frequently involve situations exhibiting an order among the different categories, e.g. a teacher always rates his/her students by giving grades on their overall performance. In contrast to metric regression problems, the grades are usually discrete and finite. These grades are also different from the class labels in classification problems due to the existence of ranking information. For example, grade labels have the ordering $F < D < C < B < A$. This is a learning task of predicting variables of ordinal scale, a setting bridging between metric regression and classification referred to as *ranking learning* or *ordinal regression*.

There is some literature about ordinal regression in the domain of machine learning. Kramer et al. (2001) investigated the use of a regression tree learner by mapping the ordinal variables into numeric values. However there might be no principled way of devising an appropriate mapping function. Frank and Hall (2001) converted an ordinal regression problem into nested binary classification problems that encode the ordering of the original ranks, and then the results of standard binary classifiers can be organized for prediction. Har-Peled et al. (2003) proposed a constraint classification approach for ranking problems based on binary classifiers. Cohen et al. (1999) considered general ranking problems in the form of preference judgements. Herbrich et al. (2000) applied the principle of Structural Risk Minimization (Vapnik, 1995) to ordinal regression leading to a new distribution-independent learning algorithm based on a loss function between pairs of ranks. Shashua and Levin (2003) generalized the formulation of support vector machines to ordinal regression and the numerical results they presented shows a significant improvement on the performance compared with the on-line algorithm proposed by Crammer and Singer (2002).

In the statistics literature, most of the approaches are based on generalized linear models (McCullagh and Nelder, 1983). The cumulative model (McCullagh, 1980) is well-known in classical statistical approaches for ordinal regression, in which they rely on a specific distributional assumption on the unobservable latent variables and a stochastic ordering of the input space. Johnson and Albert (1999) described Bayesian inference on parametric models for ordinal data using sampling techniques. Tutz (2003) presented a general framework for semiparametric models that extends generalized additive models (Hastie and Tibshirani, 1990) by incorporating nonparametric parts. The nonparametric components of the regression model are fitted by maximizing penalized log likelihood, and model selection is carried out using AIC.

Gaussian processes (O'Hagan, 1978; Neal, 1997) have provided a promising non-parametric Bayesian approach to metric regression (Williams and Rasmussen, 1996) and classification problems (Williams and Barber, 1998). The important advantage of Gaussian process models (GPs) over other non-Bayesian models is the explicit probabilistic formulation. This not only provides probabilistic predictions but also gives the ability to infer model parameters such as those that control the kernel shape and the noise level. The GPs are also different from the semiparametric approach of Tutz (2003) in several ways. First, the additive models (Fahrmeir and Tutz, 2001) are defined by functions in each input dimension, whereas the GPs can have more general non-additive covariance functions; second, the kernel trick allows to use infinite basis function expansions; third, the GPs perform Bayesian inference in the space of the latent functions.

In this paper, we present a probabilistic kernel approach to ordinal regression in Gaussian processes. We impose a Gaussian process prior distribution on the latent functions, and employ an appropriate likelihood function for ordinal variables which can be regarded as a generalization of the *probit* function. Two Bayesian inference techniques are applied to implement model adaptation by using the Laplace approximation (MacKay, 1992) and the expectation propagation (Minka, 2001) respectively. Comparisons of the generalization performance against the support vector approach (Shashua and Levin, 2003) on some benchmark and real-world data sets, such as movie ranking and gene expression analysis, verify the usefulness of this approach.

The paper is organized as follows: in section 2, we describe the Bayesian framework in Gaussian processes for ordinal regression; in section 3, we discuss the Bayesian techniques for hyperparameter inference; in section 4, we present the predictive distribution for probabilistic prediction; in section 5, we give some extensive discussion on these techniques; in section 6, we report the results of numerical experiments on some benchmark and real-world data sets; we conclude this paper in section 7.

2. Bayesian framework

Consider a data set composed of n samples. Each of the samples is a pair of input vector $x_i \in \mathcal{R}^d$ and the corresponding target $y_i \in \mathcal{Y}$ where $\mathcal{Y}$ is a finite set of r ordered categories. Without loss of generality, these categories are denoted as consecutive integers $\mathcal{Y} = \{1, 2, \ldots, r\}$ that keep the known ordering information. The main idea is to assume an unobservable latent function $f(x_i) \in \mathcal{R}$ associated with x_i in a Gaussian process, and the ordinal variable y_i dependent on the latent function $f(x_i)$ by modelling the ranks as intervals on the real line. A Bayesian framework is described with more details in the following.

2.1 Gaussian process prior

The latent functions $\{f(x_i)\}$ are usually assumed as the realizations of random variables indexed by their input vectors in a zero-mean Gaussian process. The Gaussian process can then be fully specified by giving the covariance matrix for any finite set of zero-mean random variables $\{f(x_i)\}$. The covariance between the functions corresponding to the inputs x_i and x_j can be defined by Mercer kernel functions (Wahba, 1990; Schölkopf and Smola, 2001), e.g. Gaussian kernel which is defined as

$$\mathrm{Cov}[f(x_i), f(x_j)] = \mathcal{K}(x_i, x_j) = \exp\left(-\frac{\kappa}{2}\sum_{\varsigma=1}^{d}(x_i^{\varsigma} - x_j^{\varsigma})^2\right) \tag{1}$$

where $\kappa > 0$ and x_i^{ς} denotes the ς-th element of x_i.[1] Thus, the prior probability of these latent functions $\{f(x_i)\}$ is a multivariate Gaussian

$$\mathcal{P}(\boldsymbol{f}) = \frac{1}{\mathcal{Z}_{\boldsymbol{f}}}\exp\left(-\frac{1}{2}\boldsymbol{f}^T\Sigma^{-1}\boldsymbol{f}\right) \tag{2}$$

where $\boldsymbol{f} = [f(x_1), f(x_2), \ldots, f(x_n)]^T$, $\mathcal{Z}_{\boldsymbol{f}} = (2\pi)^{\frac{n}{2}}|\Sigma|^{\frac{1}{2}}$, and Σ is the $n \times n$ covariance matrix whose ij-th element is defined as in (1).

2.2 Likelihood for ordinal variables

The likelihood is the joint probability of observing the ordinal variables given the latent functions, denoted as $\mathcal{P}(\mathcal{D}|\boldsymbol{f})$ where $\mathcal{D}$ denotes the target set $\{y_i\}$. Generally, the likelihood can be evaluated as a product of the likelihood function on individual observation:

$$\mathcal{P}(\mathcal{D}|\boldsymbol{f}) = \prod_{i=1}^{n}\mathcal{P}(y_i|f(x_i)) \tag{3}$$

where the likelihood function $\mathcal{P}(y_i|f(x_i))$ could be intuitively defined as

$$\mathcal{P}_{\mathrm{ideal}}(y_i|f(x_i)) = \begin{cases} 1 & \text{if } b_{y_i-1} < f(x_i) \le b_{y_i}, \\ 0 & \text{otherwise} \end{cases} \tag{4}$$

where $b_0 = -\infty$ and $b_r = +\infty$ are defined subsidiarily, $b_1 \in \mathcal{R}$ and the other threshold variables can be further defined as $b_j = b_1 + \sum_{\iota=2}^{j}\Delta_\iota$ with positive padding variables Δ_ι and $\iota = 2, \ldots, r-1$. The role of $b_1 < b_2 < \ldots < b_{r-1}$ is to divide the real line into

1. Other Mercer kernel functions, such as polynomial kernels and spline kernels etc., can also be used in the covariance function.

r contiguous intervals; these intervals map the real function value $f(x_i)$ into the discrete variable y_i while enforcing the ordinal constraints. The likelihood function (4) is used for ideally noise-free cases. In the presence of noise from inputs or targets, we may explicitly assume that the latent functions are contaminated by a Gaussian noise with zero mean and unknown variance σ^2.[2] $\mathcal{N}(\delta;\mu,\sigma^2)$ is used to denote a Gaussian random variable δ with mean μ and variance σ^2 henceforth. Then the ordinal likelihood function becomes

$$\mathcal{P}(y_i|f(x_i)) = \int \mathcal{P}_{\text{ideal}}(y_i|f(x_i)+\delta_i)\mathcal{N}(\delta_i;0,\sigma^2)d\delta_i = \Phi\left(z_1^i\right) - \Phi\left(z_2^i\right) \tag{5}$$

where $z_1^i = \frac{b_{y_i}-f(x_i)}{\sigma}$, $z_2^i = \frac{b_{y_i-1}-f(x_i)}{\sigma}$, and $\Phi(z) = \int_{-\infty}^{z}\mathcal{N}(\varsigma;0,1)d\varsigma$. Note that binary classification is a special case of ordinal regression when $r=2$, and in this case the likelihood function (5) becomes the *probit* function. The quantity $-\ln\mathcal{P}(y_i|f(x_i))$ is usually referred to as the loss function $\ell(y_i,f(x_i))$. The derivatives of the loss function with respect to $f(x_i)$ are needed in some approximate Bayesian inference methods. The first order derivative of the loss function can be written as

$$\frac{\partial\ell(y_i,f(x_i))}{\partial f(x_i)} = \frac{1}{\sigma}\frac{\mathcal{N}(z_1^i;0,1)-\mathcal{N}(z_2^i;0,1)}{\Phi(z_1^i)-\Phi(z_2^i)} \tag{6}$$

and the second order derivative can be given as

$$\frac{\partial^2\ell(y_i,f(x_i))}{\partial^2 f(x_i)} = \frac{1}{\sigma^2}\left(\frac{\mathcal{N}(z_1^i;0,1)-\mathcal{N}(z_2^i;0,1)}{\Phi(z_1^i)-\Phi(z_2^i)}\right)^2 + \frac{1}{\sigma^2}\frac{z_1^i\mathcal{N}(z_1^i;0,1)-z_2^i\mathcal{N}(z_2^i;0,1)}{\Phi(z_1^i)-\Phi(z_2^i)}. \tag{7}$$

We present graphs of the ordinal likelihood function (5) and the derivatives of the loss function in Figure 1 as an illustration. Note that the first order derivative (6) is a monotonically increasing function of $f(x_i)$, and the second order derivative (7) is always a positive value between 0 and $\frac{1}{\sigma^2}$. Given the facts that $\ln\mathcal{P}_{\text{ideal}}(y_i|f(x_i)+\delta_i)$ is concave in $(f(x_i),\delta_i)$ and $\ln\mathcal{N}(\delta_i;0,\sigma^2)$ is also concave, as pointed out by Pratt (1981), the convexity of the loss function follows, because the integral of a log concave function with respect to some of its arguments is a log concave function of its remaining arguments (Brascamp and Lieb, 1976, Cor. 3.5).

2.3 Posterior probability

Based on Bayes' theorem, the posterior probability can then be written as

$$\mathcal{P}(\boldsymbol{f}|\mathcal{D}) = \frac{1}{\mathcal{P}(\mathcal{D})}\prod_{i=1}^{n}\mathcal{P}(y_i|f(x_i))\,\mathcal{P}(\boldsymbol{f}) \tag{8}$$

where the prior probability $\mathcal{P}(\boldsymbol{f})$ is defined as in (2), the likelihood function $\mathcal{P}(y_i|f(x_i))$ is defined as in (5), and $\mathcal{P}(\mathcal{D}) = \int\mathcal{P}(\mathcal{D}|\boldsymbol{f})\mathcal{P}(\boldsymbol{f})d\boldsymbol{f}$.

The Bayesian framework we described above is conditional on the model parameters including the kernel parameters κ in the covariance function (1) that control the kernel shape, the threshold parameters $\{b_1,\Delta_2,\ldots,\Delta_{r-1}\}$ and the noise level σ in the likelihood function

2. In principle, any distribution rather than a Gaussian can be assumed for the noise on the latent functions.

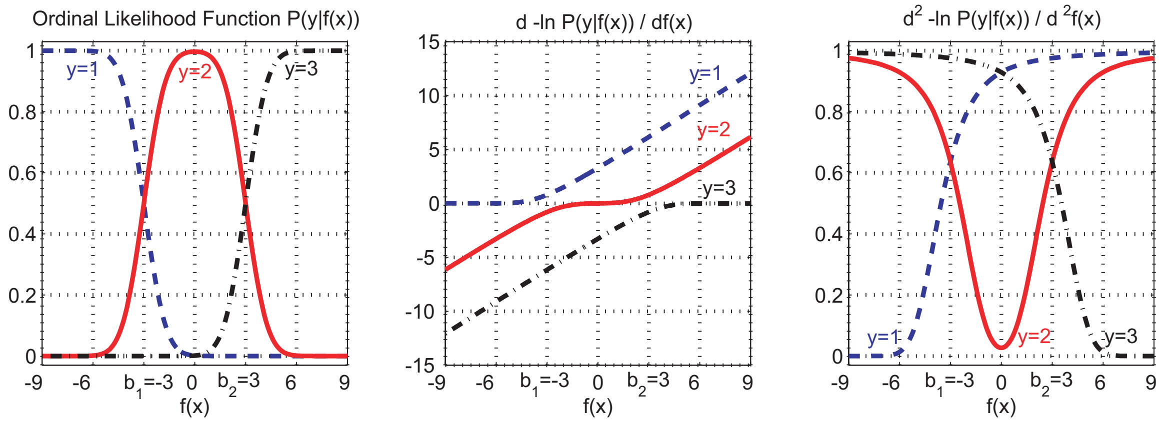

Figure 1: The graph of the likelihood function for an ordinal regression problem with $r = 3$, along with the first and second order derivatives of the loss function (negative logarithm of the likelihood function), where the noise variance $\sigma^2 = 1$, and the two thresholds are $b_1 = -3$ and $b_2 = +3$.

(5). All these parameters can be collected into $\boldsymbol{\theta}$, which is the hyperparameter vector. The normalization factor $\mathcal{P}(\mathcal{D})$ in (8), more exactly $\mathcal{P}(\mathcal{D}|\boldsymbol{\theta})$, is known as the evidence for $\boldsymbol{\theta}$, a yardstick for model selection. In the next section, we discuss techniques for hyperparameter learning.

3. Model adaptation

In a full Bayesian treatment, the hyperparameters $\boldsymbol{\theta}$ must be integrated over the $\boldsymbol{\theta}$-space. Monte Carlo methods (Neal, 1997) can be adopted here to approximate the integral effectively. However these might be prohibitively expensive to use in practice. Alternatively, we consider model selection by determining an optimal setting for $\boldsymbol{\theta}$. The optimal values of hyperparameters $\boldsymbol{\theta}$ can be *simply* inferred by maximizing the posterior probability $\mathcal{P}(\boldsymbol{\theta}|\mathcal{D})$, where $\mathcal{P}(\boldsymbol{\theta}|\mathcal{D}) \propto \mathcal{P}(\mathcal{D}|\boldsymbol{\theta})\mathcal{P}(\boldsymbol{\theta})$. The prior distribution on the hyperparameters $\mathcal{P}(\boldsymbol{\theta})$ can be specified by domain knowledge, or alternatively some vague uninformative distribution. The evidence is given by a high dimensional integral, $\mathcal{P}(\mathcal{D}|\boldsymbol{\theta}) = \int \mathcal{P}(\mathcal{D}|\boldsymbol{f})\mathcal{P}(\boldsymbol{f})\, d\boldsymbol{f}$. A popular idea for computing the evidence is to approximate the posterior distribution $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$ as a Gaussian, and then the evidence can be calculated by an explicit formula (MacKay, 1992; Csató et al., 2000; Minka, 2001). In this section, we describe two Bayesian techniques for model adaptation by using the Laplace approximation and the expectation propagation respectively.

3.1 MAP approach with Laplace approximation

The evidence can be calculated analytically after applying the Laplace approximation at the maximum a posteriori (MAP) estimate, and gradient-based optimization methods can then be used to infer the optimal hyperparameters by maximizing the evidence. The MAP estimate on the latent functions is referred to $\boldsymbol{f}_{\text{MAP}} = \arg\max_{\boldsymbol{f}} \mathcal{P}(\boldsymbol{f}|\mathcal{D})$, which is equivalent to the minimizer of negative logarithm of $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$, i.e.

$$\mathcal{S}(\boldsymbol{f}) = \sum_{i=1}^{n} \ell(y_i, f(x_i)) + \frac{1}{2}\, \boldsymbol{f}^T \Sigma^{-1} \boldsymbol{f} \tag{9}$$

where $\ell(y_i, f(x_i)) = -\ln \mathcal{P}(y_i|f(x_i))$ is known as the loss function. Note that $\frac{\partial^2 \mathcal{S}(\boldsymbol{f})}{\partial \boldsymbol{f} \partial \boldsymbol{f}^T} = \Sigma^{-1} + \Lambda$ is a positive definite matrix, where Λ is a diagonal matrix whose ii-th entry is $\frac{\partial^2 \ell(y_i, f(x_i))}{\partial^2 f(x_i)}$ given as in (7). Thus, this is a convex programming problem with a unique solution.[3] The Laplace approximation of $\mathcal{S}(\boldsymbol{f})$ refers to carrying out the Taylor expansion at the MAP point and retaining the terms up to the second order (MacKay, 1992). Since the first order derivative with respect to $\boldsymbol{f}$ vanishes at $\boldsymbol{f}_{\mathrm{MAP}}$, $\mathcal{S}(\boldsymbol{f})$ can also be written as

$$\mathcal{S}(\boldsymbol{f}) \approx \mathcal{S}(\boldsymbol{f}_{\mathrm{MAP}}) + \frac{1}{2}(\boldsymbol{f} - \boldsymbol{f}_{\mathrm{MAP}})^T \left(\Sigma^{-1} + \Lambda_{\mathrm{MAP}}\right) (\boldsymbol{f} - \boldsymbol{f}_{\mathrm{MAP}}) \tag{10}$$

where Λ_{MAP} denotes the matrix Λ at the MAP estimate. This is equivalent to approximating the posterior distribution $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$ as a Gaussian distribution centered on $\boldsymbol{f}_{\mathrm{MAP}}$ with the covariance matrix $(\Sigma^{-1} + \Lambda_{\mathrm{MAP}})^{-1}$, i.e. $\mathcal{P}(\boldsymbol{f}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{f}; \boldsymbol{f}_{\mathrm{MAP}}, (\Sigma^{-1} + \Lambda_{\mathrm{MAP}})^{-1})$. Using the Laplace approximation (10) and $\mathcal{Z}_{\boldsymbol{f}}$ defined as in (2), the evidence can be computed analytically as follows

$$\mathcal{P}(\mathcal{D}|\boldsymbol{\theta}) = \frac{1}{\mathcal{Z}_{\boldsymbol{f}}} \int \exp(-\mathcal{S}(\boldsymbol{f}))\, d\boldsymbol{f} \approx \exp(-\mathcal{S}(\boldsymbol{f}_{\mathrm{MAP}}))|\mathbf{I} + \Sigma\Lambda_{\mathrm{MAP}}|^{-\frac{1}{2}} \tag{11}$$

where $\mathbf{I}$ is an $n \times n$ identity matrix. The gradients of the logarithm of the evidence (11) with respect to the hyperparameters $\boldsymbol{\theta}$ can be derived analytically. Then gradient-based optimization methods can be employed to search for the maximizer of the evidence. Refer to Appendix A for the detailed gradient formulae and the outline of our algorithm for model adaptation.

3.2 Expectation propagation with variational methods

The expectation propagation algorithm (EP) is an approximate Bayesian inference method (Minka, 2001), which can be regarded as an extension of assumed-density-filter (ADF). The EP algorithm has been applied in Gaussian process classification along with variational methods for model selection (Seeger, 2002; Kim and Ghahramani, 2003). In the setting of Gaussian processes, EP attempts to approximate $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$ as a product distribution in the form of $Q(\boldsymbol{f}) = \prod_{i=1}^{n} \tilde{t}_i(f(x_i))\mathcal{P}(\boldsymbol{f})$ where $\tilde{t}_i(f(x_i)) = s_i \exp(-\frac{1}{2}p_i(f(x_i) - m_i)^2)$. The parameters $\{s_i, m_i, p_i\}$ in $\{\tilde{t}_i\}$ are successively optimized by minimizing the following Kullback-Leibler divergence,

$$\tilde{t}_i^{\mathrm{new}} = \arg\min_{\tilde{t}_i} \mathbf{KL}\left(\frac{Q(\boldsymbol{f})}{\tilde{t}_i^{\mathrm{old}}}\mathcal{P}(y_i|f(x_i)) \,\middle\|\, \frac{Q(\boldsymbol{f})}{\tilde{t}_i^{\mathrm{old}}}\tilde{t}_i\right). \tag{12}$$

Since $Q(\boldsymbol{f})$ is in the exponential family, this minimization can be simply solved by moment matching up to the second order. A detailed updating scheme can be found in Appendix B.

3. The Newton-Raphson formula can be used to find the solution for simple cases.

At the equilibrium of $Q(\boldsymbol{f})$, we obtain an approximate posterior distribution as $\mathcal{P}(\boldsymbol{f}|\mathcal{D}) \approx \mathcal{N}(\boldsymbol{f}; (\Sigma^{-1}+\Pi)^{-1}\Pi\boldsymbol{m}, (\Sigma^{-1}+\Pi)^{-1})$ where Π is a diagonal matrix whose ii-th entry is p_i and $\boldsymbol{m} = [m_1, m_2, \ldots, m_n]^T$.

Variational methods can be used to optimize the hyperparameters $\boldsymbol{\theta}$ by maximizing the lower bound on the logarithm of the evidence. By applying Jensen's inequality, we have

$$\begin{aligned} \log\mathcal{P}(\mathcal{D}|\boldsymbol{\theta}) &= \log\int \tfrac{\mathcal{P}(\mathcal{D}|\boldsymbol{f})\mathcal{P}(\boldsymbol{f})}{Q(\boldsymbol{f})} Q(\boldsymbol{f}) d\boldsymbol{f} \geq \int Q(\boldsymbol{f}) \log \tfrac{\mathcal{P}(\mathcal{D}|\boldsymbol{f})\mathcal{P}(\boldsymbol{f})}{Q(\boldsymbol{f})} d\boldsymbol{f} \\ &= \int Q(\boldsymbol{f}) \log\mathcal{P}(\mathcal{D}|\boldsymbol{f}) d\boldsymbol{f} + \int Q(\boldsymbol{f}) \log\mathcal{P}(\boldsymbol{f}) d\boldsymbol{f} - \int Q(\boldsymbol{f}) \log Q(\boldsymbol{f}) d\boldsymbol{f} = \mathcal{F}(\boldsymbol{\theta}). \end{aligned} \tag{13}$$

The lower bound $\mathcal{F}(\boldsymbol{\theta})$ can be written as an explicit expression at the equilibrium of $Q(\boldsymbol{f})$, and then the gradients with respect to $\boldsymbol{\theta}$ can be derived by neglecting the possible dependency of $Q(\boldsymbol{f})$ on $\boldsymbol{\theta}$. The detailed formulation can be found in Appendix C.

4. Prediction

We have described two techniques, the MAP approach and the EP approach, to infer the optimal model. At the optimal hyperparameters we inferred, denoted as $\boldsymbol{\theta}^*$, let us take a test case x for which the target y_x is unknown. The latent variable $f(x)$ and the column vector $\boldsymbol{f}$ containing the n zero-mean random variables $\{f(x_i)\}_{i=1}^n$ have the prior joint multivariate Gaussian distribution, i.e.

$$\left[\begin{array}{c} \boldsymbol{f} \\ f(x) \end{array}\right] \sim \mathcal{N}\left[\left(\begin{array}{c} 0 \\ 0 \end{array}\right), \left(\begin{array}{cc} \Sigma & \boldsymbol{k} \\ \boldsymbol{k}^T & \mathcal{K}(x,x) \end{array}\right)\right]$$

where $\boldsymbol{k} = [\mathcal{K}(x,x_1), \mathcal{K}(x,x_2), \ldots, \mathcal{K}(x,x_n)]^T$. The conditional distribution of $f(x)$ given $\boldsymbol{f}$ is a Gaussian too, denoted as $\mathcal{P}(f(x)|\boldsymbol{f},\boldsymbol{\theta}^*)$ with mean $\boldsymbol{f}^T\Sigma^{-1}\boldsymbol{k}$ and variance $\mathcal{K}(x,x) - \boldsymbol{k}^T\Sigma^{-1}\boldsymbol{k}$. The predictive distribution of $\mathcal{P}(f(x)|\mathcal{D},\boldsymbol{\theta}^*)$ can be computed as an integral over $\boldsymbol{f}$-space, which can be written as

$$\mathcal{P}(f(x)|\mathcal{D},\boldsymbol{\theta}^*) = \int \mathcal{P}(f(x)|\boldsymbol{f},\boldsymbol{\theta}^*)\mathcal{P}(\boldsymbol{f}\,|\mathcal{D},\boldsymbol{\theta}^*)\, d\boldsymbol{f}. \tag{14}$$

The posterior distribution $\mathcal{P}(\boldsymbol{f}|\mathcal{D},\boldsymbol{\theta}^*)$ can be approximated as a Gaussian by the MAP approach or the EP approach (refer to Section 3). The predictive distribution (14) can then be simplified as a Gaussian $\mathcal{N}(f(x);\mu_x,\sigma_x^2)$ with mean μ_x and variance σ_x^2. In the MAP approach, we reach

$$\mu_x = \boldsymbol{k}^T\Sigma^{-1}\boldsymbol{f}_{\mathrm{MAP}} \quad \text{and} \quad \sigma_x^2 = \mathcal{K}(x,x) - \boldsymbol{k}^T(\Sigma + \Lambda_{\mathrm{MAP}}^{-1})^{-1}\boldsymbol{k}. \tag{15}$$

While in the EP approach, we get

$$\mu_x = \boldsymbol{k}^T(\Sigma + \Pi^{-1})^{-1}\boldsymbol{m} \quad \text{and} \quad \sigma_x^2 = \mathcal{K}(x,x) - \boldsymbol{k}^T(\Sigma + \Pi^{-1})^{-1}\boldsymbol{k}. \tag{16}$$

The predictive distribution over ordinal targets y_x is

$$\begin{aligned} \mathcal{P}(y_x|x,\mathcal{D},\boldsymbol{\theta}^*) &= \int \mathcal{P}(y_x|f(x),\boldsymbol{\theta}^*)\mathcal{P}(f(x)|\mathcal{D},\boldsymbol{\theta}^*)\, df(x) \\ &= \Phi\left(\frac{b_{y_x}-\mu_x}{\sqrt{\sigma^2+\sigma_x^2}}\right) - \Phi\left(\frac{b_{y_x-1}-\mu_x}{\sqrt{\sigma^2+\sigma_x^2}}\right). \end{aligned}$$

The predictive ordinal scale can be decided as $\arg\max_i \mathcal{P}(y_x = i|x,\mathcal{D},\boldsymbol{\theta}^*)$.

5. Discussion

In the MAP approach, the mean of the predictive distribution depends on the MAP estimate $\boldsymbol{f}_{\text{MAP}}$, which is unique and can be found by solving a convex programming problem. Evidence maximization is useful if the Laplace approximation around the mode point $\boldsymbol{f}_{\text{MAP}}$ gives a good summary of the posterior distribution $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$. While in the approach of expectation propagation, the mean of the predictive distribution depends on the approximate mean of the posterior distribution. When the true shape of $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$ is far from a Gaussian centered on the mode, the EP approach can have a great advantage over the Laplace approximation. However the EP algorithm cannot guarantee convergence, though it usually works well in practice.

The gradient-based optimization method usually requests evidence evaluation at tens of different settings of $\boldsymbol{\theta}$ before the minimum is found. For each $\boldsymbol{\theta}$, the inversion of the matrix Σ is required that costs time at $\mathcal{O}(n^3)$, where n is the number of training samples. Recently, Csató and Opper (2002) proposed a fast training algorithm for Gaussian processes in which the set of basis vectors are determined on-line for sparse representation. Lawrence et al. (2003) proposed a greedy selection with criteria based on information-theoretic principles for sparse Gaussian processes (Seeger, 2003). Tresp (2000) proposed the Bayesian committee machines to divide and conquer large datasets, while using infinite mixtures of Gaussian Processes (Rasmussen and Ghahramani, 2002) is another promising technique. These algorithms can be applied directly in the settings of ordinal regression for speedup.

Feature selection is an essential part in modelling. In Gaussian processes, the automatic relevance determination (ARD) method proposed by MacKay (1994) and Neal (1996) can be embedded into the covariance function (1) as follows:

$$Cov[f(x_i), f(x_j)] = \mathcal{K}(x_i, x_j) = \exp\left(-\frac{1}{2}\sum_{\varsigma=1}^{d} \kappa_\varsigma (x_i^\varsigma - x_j^\varsigma)^2\right) \tag{17}$$

where $\kappa_\varsigma > 0$ is the ARD parameter.[4] The gradients with respect to the variables $\{\ln \kappa_\varsigma\}$ can also be derived analytically for model adaptation. The optimal value of the ARD parameter κ_ς indicates the relevance of the ς-th input feature to the target. The form of feature selection we use here results in a type of feature weighting. Furthermore, the linear combination of heterogeneous kernels with positive coefficients is still a valid covariance function. Lanckriet et al. (2004) suggest to learn the kernel matrix with semidefinite programming. In the Bayesian framework, these positive coefficients for kernels could be treated as hyperparameters, and optimized using the evidence as a criterion for optimization.

Note that binary classification is a special case of ordinal regression with $r = 2$, and the likelihood function (5) becomes the *probit* function when $r = 2$. Both of the *probit* function and the logistic function can be used as the likelihood function in binary classification, while they have different origins. Due to the dichotomous nature in the classes of multi-classification, discriminant functions are constructed for each class and then compete again others via the *softmax* function to determine the likelihood. The logistic function, as a special case of the *softmax* function, comes from general classification problems.

4. These ARD parameters control the covariance length-scale of the Gaussian process along each input dimension.

In metric regression, warped Gaussian processes (Snelson et al., 2004) assume that there is a nonlinear, monotonic, and continuous warping function relating the observed targets and some latent variables in a Gaussian process. The warping function, which is learned from the data, can be thought of as a pre-processing transformation applied before modelling with a Gaussian process. A different (and very common) approach to dealing with this preprocessing is to *discretize* the target values into r different bins. These discrete values are clearly ordinal, and applying ordinal regression to these discrete values seems the natural choice. Interestingly, as the number of discretization bins r is increased, the ordinal regression model becomes very similar to the warped Gaussian processes model. In particular, by varying the thresholds in our ordinal regression model, it can approximate any continuous warping function.

6. Numerical experiments

We start this section with a simple synthetic dataset to visualize the behavior of these algorithms, and report the experimental results on sixteen benchmark datasets.[5] Then we perform experiments on a collaborative filtering problem using the "EachMovie" data, and on Gleason score prediction from gene microarray data related to prostate cancer. Shashua and Levin (2003) generalized the support vector formulation by finding multiple thresholds to define parallel discriminant hyperplanes for ordinal scales, and reported that the performance of the support vector approach is better than that of the on-line algorithm (Crammer and Singer, 2002). The problem size in the large-margin ranking algorithm of Herbrich et al. (2000) is a quadratic function of the training data size making the algorithmic complexity $\mathcal{O}(n^4)$–$\mathcal{O}(n^6)$. This makes the experiments on large datasets computationally difficult. Thus, we decide to limit our comparisons to the support vector approach (SVM) of Shashua and Levin (2003) and the two versions of our approach, the MAP approach with Laplace approximation (MAP) and the EP algorithm with variational methods (EP). In our implementation,[6] we used the routine L-BFGS-B (Byrd et al., 1995) as the gradient-based optimization package, and started from the initial values of hyperparameters to infer the optimal values in the criterion of the approximate evidence (11) for MAP or the variational lower bound (13) for EP respectively.[7] The improved SMO algorithm (Keerthi et al., 2001) was adapted to implement the SVM approach (refer to Chu and Keerthi (2005) for detailed description and extensive discussion),[8] and 5-fold cross validation was used to determine the optimal values of model parameters (the kernel parameter κ and the regularization factor C) involved in the problem formulations. The initial search was done on a 7×7 coarse grid linearly spaced in the region $\{(\log_{10} C, \log_{10} \kappa) | -3 \leq \log_{10} C \leq 3, -3 \leq \log_{10} \kappa \leq 3\}$, followed by a fine search on a 9×9 uniform grid linearly spaced by 0.2 in the $(\log_{10} C, \log_{10} \kappa)$ space. We have utilized two evaluation metrics which quantify the accuracy of predictive ordinal scales $\{\hat{y}_1, \ldots, \hat{y}_t\}$ with respect to true targets $\{y_1, \ldots, y_t\}$:

5. These datasets are publicly available at http://www.liacc.up.pt/∼ltorgo/Regression/DataSets.html.
6. The two versions of our proposed approach were implemented in ANSI C, and the source code is accessible at http://www.gatsby.ucl.ac.uk/∼chuwei/code/gpor.tar.
7. In numerical experiments, the initial values of the hyperparameters were usually chosen as $\sigma^2 = 1$, $\kappa = 1/d$ for Gaussian kernel, the threshold $b_1 = -1$ and $\Delta_\iota = 2/r$. We suggest to try several starting points in practice, and then choose the best model by the objective functional.
8. The source code in ANSI C is available at http://www.gatsby.ucl.ac.uk/∼chuwei/code/svorim.tar.

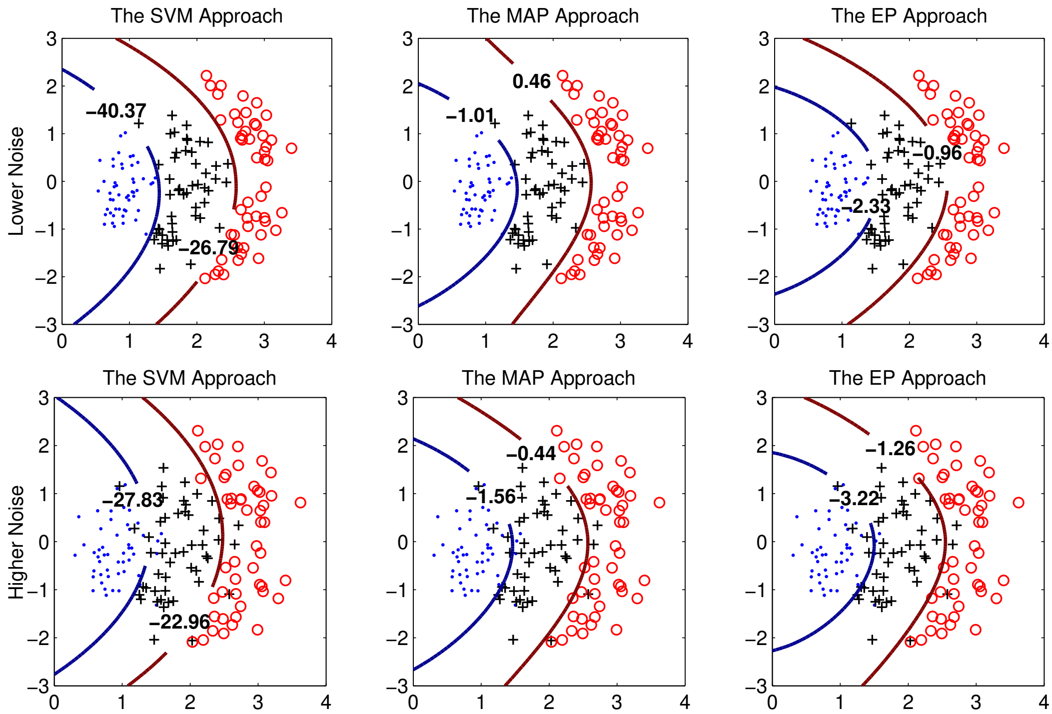

Figure 2: The performance of the three algorithms on a synthetic three-rank ordinal regression problem. The discriminant function values of the SVM approach, and the predictive mean values of the two Gaussian process approaches are presented as contour graphs indexed by the two thresholds. The upper graphs are for the case of lower noise level, while the lower graphs are for the case of higher noise level. The training samples we used are presented in these graphs. The dots denote the training samples of rank 1, the crosses denote the training samples of rank 2 and the circles denote the training samples of rank 3.

- *Mean absolute error* is the average deviation of the prediction from the true target, i.e. $\frac{1}{t}\sum_{i=1}^{t}|\hat{y}_i - y_i|$, in which we treat the ordinal scales as consecutive integers;
- *Mean zero-one error* gives an error of 1 to every incorrect prediction that is the fraction of incorrect predictions.

6.1 Artificial data

Figure 2 presents the behavior of the three algorithms using the Gaussian kernel (1) on a synthetic 2D data with three ordinal scales. In the support vector approach, the optimal thresholds were determined by the SMO algorithm and 5-fold cross validation was used to decide the optimal values of the kernel parameter and the regularization factor. As for the Gaussian process algorithms, model adaptation (see Section 3) was used to determine the

Table 1: Datasets and their characteristics. "Attributes" state the number of numerical and nominal attributes. "Training Instances" and "Instances for Test" specify the size of training/test partition. The partitions we generated and the test results on individual partitions can be accessed at http://www.gatsby.ucl.ac.uk/∼chuwei/ordinalregression.html.

	Datasets	Attributes(Numeric,Nominal)	Training Instances	Instances for Test
Set I	Diabetes	2(2,0)	30	13
	Pyrimidines	27(27,0)	50	24
	Triazines	60(60,0)	100	86
	Wisconsin Breast Cancer	32(32,0)	130	64
	Machine CPU	6(6,0)	150	59
	Auto MPG	7(4,3)	200	192
	Boston Housing	13(12,1)	300	206
	Stocks Domain	9(9,0)	600	350
	Abalone	8(7,1)	1000	3177
Set II	Bank Domains(1)	8(8,0)	50	8142
	Bank Domains(2)	32(32,0)	75	8117
	Computer Activity(1)	12(12,0)	100	8092
	Computer Activity(2)	21(21,0)	125	8067
	California Housing	8(8,0)	150	15490
	Census Domains(1)	8(8,0)	175	16609
	Census Domains(2)	16(16,0)	200	16584

optimal values of the kernel parameter, the noise level and the thresholds automatically. The figure shows that all the algorithms are working reasonably well on this task.

6.2 Benchmark data

We collected nine benchmark datasets (Set I in Table 1) that were used for metric regression problems. The target values were discretized into ordinal quantities using equal-length binning. These bins divide the range of target values into a given number of intervals that are of same length. The resulting rank values are ordered, representing these intervals of the original metric quantities. For each dataset, we generated two versions by discretizing the target values into five and ten intervals respectively. We randomly partitioned each dataset into training/test splits as specified in Table 1. The partition was repeated 20 times independently. The Gaussian kernel (1) was used in these three algorithms. The test results are recorded in Table 2 and 3. The performance of the MAP and EP approaches are closely matching. Our Gaussian process algorithms often yield better results than the support vector approach on the average value, especially when the number of training samples is small.

In the next experiment, we selected seven very large metric regression datasets (Set II in Table 1). The input vectors were normalized to zero mean and unit variance coordinate-wise. The target values of these datasets were discretized into 10 ordinal quantities using equal-frequency binning. For each dataset, a small subset was randomly selected for training and then tested on the remaining samples, as specified in Table 1. The partition was repeated 100 times independently. To show the advantage of explicitly modelling the ordinal nature of the targets, we also employed the standard Gaussian process algorithm (Williams and

Table 2: Test results of the three algorithms using a Gaussian kernel. The targets of these benchmark datasets were discretized by 5 equal-length bins. The results are the averages over 20 trials, along with the standard deviation. We use the bold face to indicate the cases in which the average value is the lowest in the results of the three algorithms. The symbols $\star$ are used to indicate the cases in which the indicated entry is significantly worse than the winning entry; A p-value threshold of 0.01 in Wilcoxon rank sum test was used to decide statistical significance.

	Mean zero-one error			Mean absolute error		
Data	SVM	MAP	EP	SVM	MAP	EP
Diabetes	57.31±12.09%	**54.23±13.78%**	54.23±13.78%	0.7462±0.1414	**0.6615±0.1376**	0.6654±0.1373
Pyrimidines	41.46±8.49%	39.79±7.21%	**36.46±6.47%**	0.4500±0.1136	0.4271±0.0906	**0.3917±0.0745**
Triazines	54.19±1.48%	52.91±2.15%	**52.62±2.66%**	0.6977±0.0259	**0.6872±0.0229**	0.6878±0.0295
Wisconsin	*70.78±3.73%	**65.00±4.71%**	65.16±4.65%	1.0031±0.0727	**1.0102±0.0937**	1.0141±0.0932
Machine	17.37±3.56%	**16.53±3.56%**	16.78±3.88%	0.1915±0.0423	**0.1847±0.0404**	0.1856±0.0424
Auto MPG	*25.73±2.24%	23.78±1.85%	**23.75±1.74%**	0.2596±0.0230	0.2411±0.0189	**0.2411±0.0186**
Boston	25.56±1.98%	24.88±2.02%	**24.49±1.85%**	0.2672±0.0190	0.2604±0.0206	**0.2585±0.0200**
Stocks	**10.81±1.70%**	11.99±2.34%	12.00±2.06%	**0.1081±0.0170**	0.1199±0.0234	0.1200±0.0206
Abalone	21.58±0.32%	**21.50±0.22%**	21.56±0.36%	**0.2293±0.0038**	0.2322±0.0025	*0.2337±0.0072

Table 3: Test results of the three algorithms using a Gaussian kernel. The targets of these benchmark datasets were discretized by 10 equal-length bins. The results are the averages over 20 trials, along with the standard deviation. We use the bold face to indicate the cases in which the average value is the lowest in the results of the three algorithms. The symbols $\star$ are used to indicate the cases in which the indicated entry is significantly worse than the winning entry; A p-value threshold of 0.01 in Wilcoxon rank sum test was used to decide statistical significance.

	Mean zero-one error			Mean absolute error		
Data	SVM	MAP	EP	SVM	MAP	EP
Diabetes	*90.38±7.00%	83.46±5.73%	**83.08±5.91%**	2.4577±0.4369	**2.1385±0.3317**	2.1423±0.3314
Pyrimidines	59.37±7.63%	55.42±8.01%	**54.38±7.70%**	0.9187±0.1895	0.8771±0.1749	**0.8292±0.1338**
Triazines	*67.91±3.63%	**63.72±4.34%**	64.01±3.78%	1.2308±0.0874	**1.1994±0.0671**	1.2012±0.0680
Wisconsin	*85.86±3.78%	**78.52±3.58%**	78.52±3.51%	2.1250±0.1500	**2.1391±0.1797**	2.1437±0.1790
Machine	**32.63±3.84%**	33.81±3.91%	33.73±3.64%	**0.4398±0.0688**	0.4746±0.0727	0.4686±0.0763
Auto MPG	44.01±2.30%	43.96±2.81%	**43.88±2.60%**	0.5081±0.0263	0.4990±0.0352	**0.4979±0.0340**
Boston	42.06±2.49%	41.53±2.77%	**41.26±2.86%**	0.4971±0.0305	0.4920±0.0330	**0.4896±0.0346**
Stocks	**17.74±2.15%**	*19.90±1.72%	*19.44±1.91%	**0.1804±0.0213**	*0.2006±0.0166	*0.1960±0.0184
Abalone	42.84±0.86%	42.60±0.91%	**42.27±0.46%**	0.5160±0.0087	0.5140±0.0075	**0.5113±0.0053**

Rasmussen, 1996) for metric regression (GPR)[9] to tackle these ordinal regression tasks, where the ordinal targets were naively treated as continuous values and the predictions for test cases were rounded to the nearest ordinal scale. The Gaussian kernel (1) was used in the four algorithms. From the test results in Table 4, the ordinal regression algorithms are clearly superior to the naive approach of applying standard metric regression. We also observed that the performance of Gaussian process algorithms are significantly better than that of the support vector approach on six of the seven datasets. This verifies our judgement in the previous experiment that our Gaussian process algorithms yield better performance

9. In the GPR, the type-II maximum likelihood was used for model selection.

Table 4: Test results of the four algorithms using a Gaussian kernel. The targets of these benchmark datasets were discretized by 10 equal-frequency bins. The results are the average over 100 trials, along with the standard deviation. "GPR" denotes the standard algorithm of Gaussian process metric regression that treats the ordinal scales as continuous values. "NLL" denotes the negative logarithm of the likelihood in prediction. We use the bold face to indicate the cases in which the average value is the lowest mean zero-one error of the four algorithms. The symbols $\star$ are used to indicate the cases in which the indicated entry is significantly worse than the winning entry; A p-value threshold of 0.01 in Wilcoxon rank sum test was used to decide statistical significance.

	Mean zero-one error				NLL	
Data	GPR	SVM	MAP	EP	MAP	EP
Bank(1)	*59.43 ± 2.80 %	49.07 ± 2.69 %	48.65 ± 1.93 %	**48.35 ± 1.91 %**	1.14 ± 0.07	1.14 ± 0.07
Bank(2)	*86.37 ± 1.49 %	*82.26 ± 2.06 %	80.96 ± 1.51 %	**80.89 ± 1.52 %**	2.20 ± 0.09	2.20 ± 0.09
CompAct(1)	*65.52 ± 2.31 %	*59.87 ± 2.25 %	58.52 ± 1.73 %	**58.51 ± 1.53 %**	1.65 ± 0.16	1.64 ± 0.14
CompAct(2)	*59.30 ± 2.27 %	*54.79 ± 2.10 %	**53.80 ± 1.84 %**	53.92 ± 1.68 %	1.49 ± 0.11	1.48 ± 0.09
California	*76.13 ± 1.27 %	*70.63 ± 1.40 %	69.60 ± 1.12 %	**69.58 ± 1.11 %**	1.89 ± 0.08	1.89 ± 0.09
Census(1)	*78.06 ± 0.81 %	*74.69 ± 0.94 %	**73.71 ± 0.77 %**	73.71 ± 0.77 %	2.04 ± 0.08	2.05 ± 0.08
Census(2)	*78.02 ± 0.85 %	*76.01 ± 1.03 %	74.53 ± 0.81 %	**74.48 ± 0.84 %**	2.03 ± 0.06	2.03 ± 0.07

than the support vector approach on small datasets. Although the EP approach often yields better results of mean zero-one error than the MAP approach on these tasks, we have not detected any statistically significant difference on their performance. In Table 4 we also report their negative logarithm of the likelihood in prediction (NLL). The performance of the MAP and EP approaches are closely matching too with no statistically significant difference.

For these datasets, the overall training time of MAP and EP approaches was substantially less than that of the SVM approach. This is because the MAP and EP approaches can tune the model parameters by gradient descent that usually required evidence evaluations at tens of different settings of $\boldsymbol{\theta}$, whereas k-fold cross validation for the SVM approach required evaluations at 130 different nodes of $\boldsymbol{\theta}$ on the grid for every fold. For larger data sets, the SVM approach may still have an advantage on training time due to the sparseness property in its computation.

6.3 Collaborative filtering

Collaborative filtering exploits correlations between ratings across a population of users. The goal is to predict a person's rating on new items given the person's past ratings on similar items and the ratings of other people on all the items (including the new item). The ratings are ordered, making collaborative filtering an ordinal regression problem. We carried out ordinal regression on a subset of the EachMovie data (Compaq, 2001).[10] The rates given by the user with ID number "52647" on 449 movies were used as the targets, in which the numbers of zero-to-five star are 40, 20, 57, 113, 145 and 74 respectively. We selected 1500 users who contributed the most ratings on these 449 movies as the input

10. The Compaq System Research Center ran the EachMovie service for 18 months. 72916 users entered a total of 2811983 numeric ratings on 1628 movies, i.e. about 2.4% are rated by zero-to-five star.

features. The ratings given by the 1500 users on each movie were used as the input vector accordingly. In the 449×1500 input matrix, about 40% elements were observed. We randomly selected a subset with size $\{50, 100, \ldots, 300\}$ of the 449 movies for training, and then tested on the remaining movies. At each size, the random selection was carried out 20 times independently.

Pearson correlation coefficient is the most popular correlation measure (Basilico and Hofmann, 2004), which corresponds to a dot product between normalized rating vectors. For instance, if applied to the movies, we can define the so-called z-scores as

$$z(v, u) = \frac{r(v, u) - \mu(v)}{\sigma(v)}$$

where u indexes users, v indexes movies, and $r(v, u)$ is the rating on the movie v given by the user u. $\mu(v)$ and $\sigma(v)$ are the movie-specific mean and standard deviation respectively. This correlation coefficient, defined as

$$\mathcal{K}(v, v') = \sum_u z(v, u) z(v', u)$$

where $\sum_u$ denotes summing over all the users, was used as the covariance/kernel function in our experiments for the three algorithms. As not all ratings are observed in the input vectors, we consider two *ad hoc* strategies to deal with missing values: mean imputation and weighted low-rank approximation. In the first case, unobserved values are identified with the mean value, that means their corresponding z-score is zero. In the second case, we applied the EM procedure described by Srebro and Jaakkola (2003) to fill in the missing data with the estimate. In the input matrix, observed elements were weighted by one and missing data were given weight zero. The low rank was fixed at 2. In Figure 3, we present the test results of the two cases at different training data size. Using mean imputation, SVM produced a bit more accurate results than Gaussian processes on mean absolute error. In the cases with low rank approximation as preprocessing, the performance of the three algorithms are highly competitive, and more interestingly, we observed about 0.08 improvement on mean absolute error for all the three algorithms. A serious treatment on the missing data could be an interesting research topic for future work.

6.4 Gene expression analysis

Singh et al. (2002) carried out microarray expression analysis on 12600 genes to identify genes that might anticipate the clinical behavior of prostate cancer. Fifty-two samples of prostate tumor were investigated. For each sample, the Gleason score ranging from 6 to 10, was given by the pathologist reflecting the level of differentiation of the glands in the prostate tumor. Predicting the Gleason score from the gene expression data is thus a typical ordinal regression problem. Since only 6 samples had a score greater than 7, we merged them as the top level, leading to three levels $\{=6, =7, \geq 8\}$ with 26, 20 and 6 samples respectively. We randomly partitioned the data into 2 folds for training and test and repeated this partitioning 20 times independently. An ARD linear kernel $\mathcal{K}(x_i, x_j) = \sum_{\varsigma=1}^{d} \kappa_\varsigma x_i^\varsigma x_j^\varsigma$ was used to evaluate feature relevance. These ARD parameters $\{\kappa_\varsigma\}$ were optimized by evidence maximization. According to the optimal values of these ARD parameters, the genes were

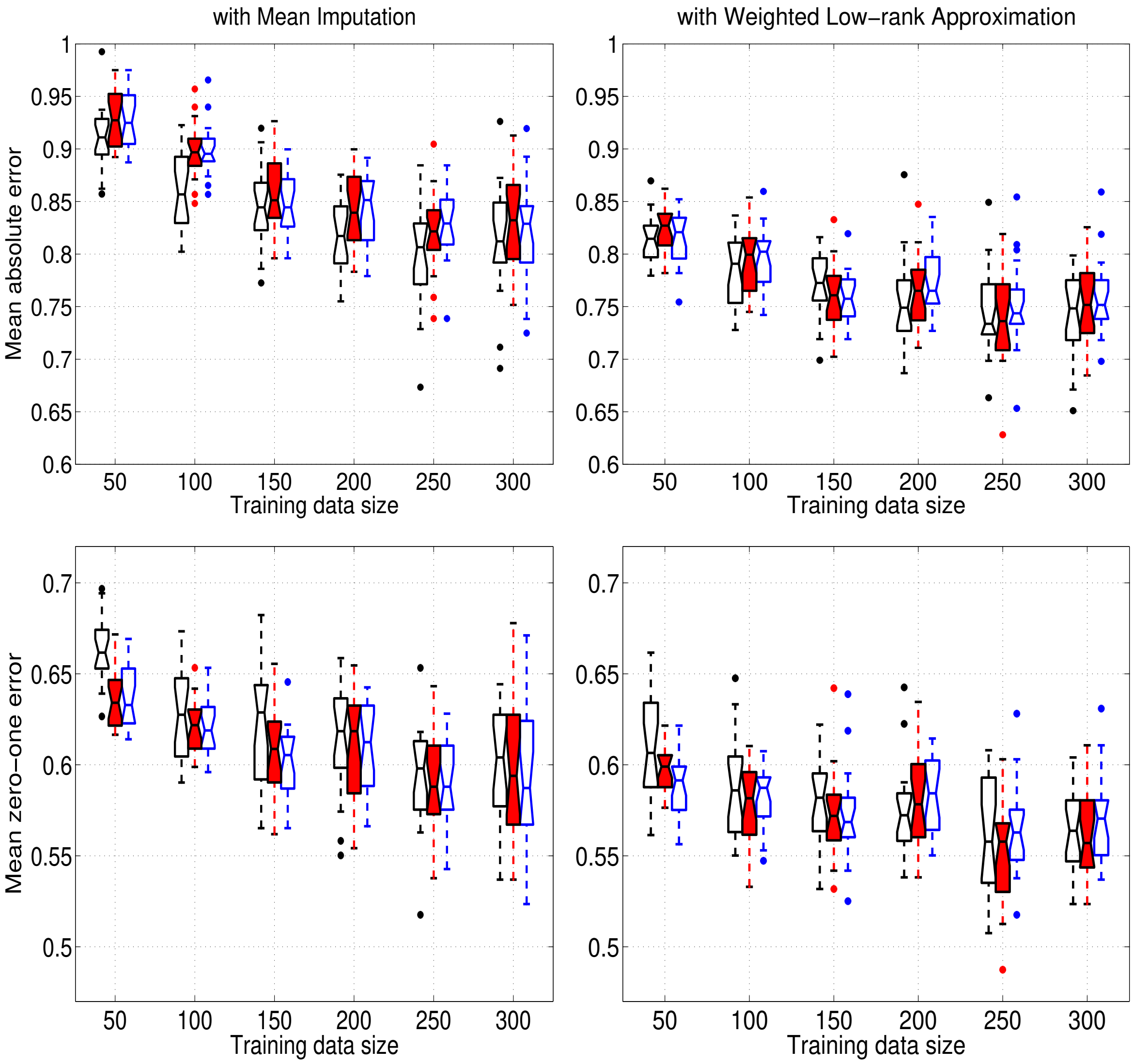

Figure 3: The test results of the three algorithms on the subset of EachMovie data over 20 trials. The grouped boxes represent the results of SVM (left), MAP (middle) and EP (right) respectively at different training data size. The notched-boxes have lines at the lower quartile, median, and upper quartile values. The whiskers are lines extending from each end of the box to the most extreme data value within 1.5·IQR(Interquartile Range) of the box. Outliers are data with values beyond the ends of the whiskers, which are displayed by dots. The higher graphs are for the results of mean absolute error and the lower graphs are for mean zero-one error. The cases of mean imputation are presented in the left graphs, and the cases with weighted low-rank approximation as preprocessing are presented in the right graphs.

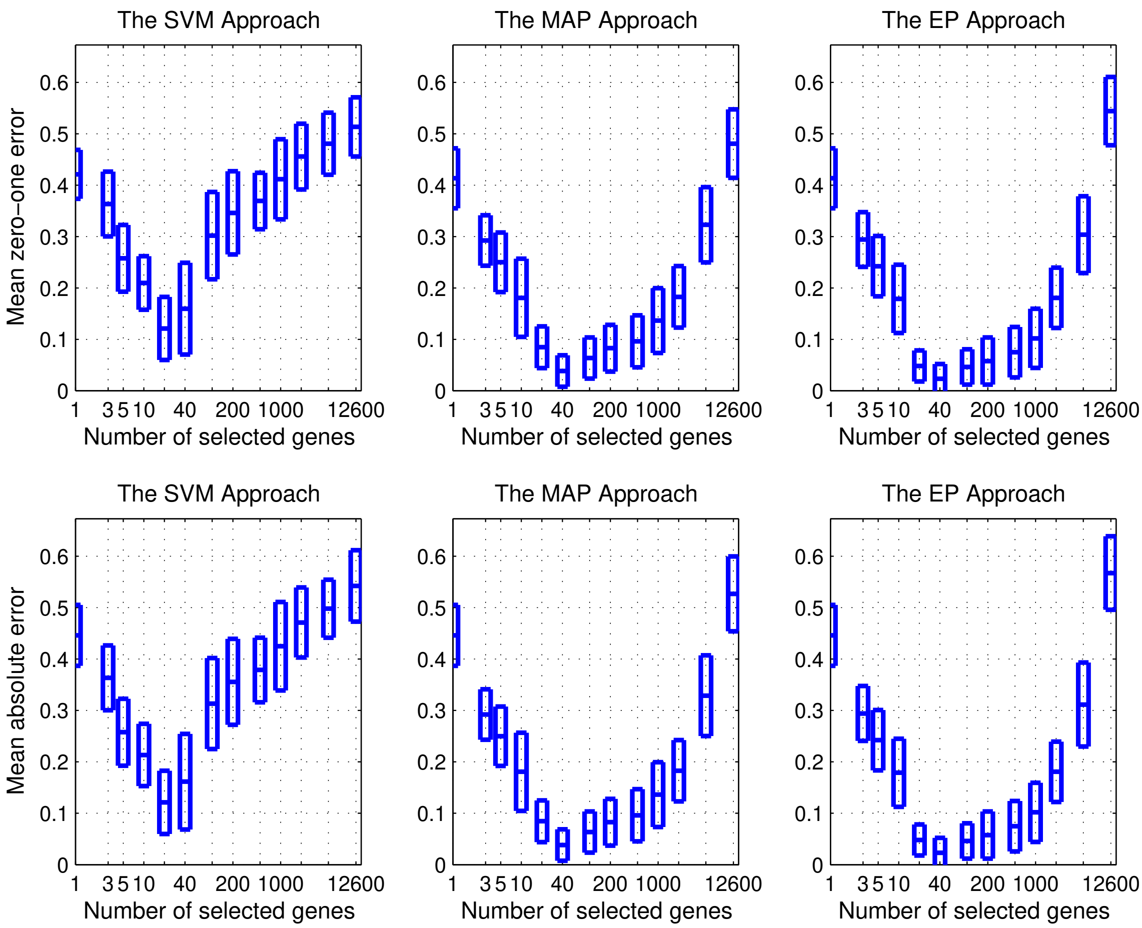

Figure 4: The test results of the three algorithms using a linear kernel on the prostate cancer data of selected genes. The horizonal axes are indexed on $\log_2$ scale. The rungs in these boxes indicate the mean values, and the heights of these vertical boxes indicate the standard deviations over the 20 trials.

ranked from irrelevant to relevant. We then removed the irrelevant genes gradually based on the rank list. The gene number was reduced from 12600 to 1. At each number of selected genes, a linear kernel $\mathcal{K}(x_i, x_j) = \sum_{\varsigma=1}^{d} x_i^\varsigma x_j^\varsigma$ was used in the three algorithms for a fair comparison. Figure 4 presents the test results of the three algorithms for different numbers of selected genes. We observed great and steady improvement using the subset of genes selected by the ARD technique. The best validation output is achieved around 40 top-ranked features. In this case, with only 26 training samples, the Bayesian approaches perform much better than the SVM, and the EP approach is generally better than the MAP approach but the difference is not statistically significant.

7. Conclusion

Ordinal regression is an important supervised learning problem with properties of both metric regression and classification. In this paper, we proposed a simple yet novel nonparametric Bayesian approach to ordinal regression based on a generalization of the *probit* likelihood function for Gaussian processes. Two approximate inference procedures were derived in detail for evidence evaluation and model adaptation. The approach intrinsically incorporates ARD feature selection and provides probabilistic prediction. The existent fast algorithms for Gaussian processes can be adapted directly to tackle relatively large datasets. Experiments on benchmark and real-world data sets show that the generalization performance is competitive and often better than support vector methods.

Acknowledgments

The main part of this work was carried out at Institute for Pure and Applied Mathematics (IPAM) of UCLA. We thank David L. Wild for stimulating this work and for many discussions. We also thank David J. C. MacKay for valuable comments. Wei Chu was supported by the National Institutes of Health and its National Institute of General Medical Sciences division under Grant Number 1 P01 GM63208. Zoubin Ghahramani was partially supported from CMU by DARPA under the CALO project. The reviewers' thoughtful comments are gratefully appreciated.

Appendix A. Gradient formulae for evidence maximization

Evidence maximization is equivalent to finding the minimizer of the negative logarithm of the evidence which can be written in an explicit expression as follows

$$-\ln \mathcal{P}(\mathcal{D}|\boldsymbol{\theta}) \approx \sum_{i=1}^{n} \ell(y_i, f_{\mathrm{MAP}}(x_i)) + \frac{1}{2}\boldsymbol{f}_{\mathrm{MAP}}^T \Sigma^{-1} \boldsymbol{f}_{\mathrm{MAP}} + \frac{1}{2}\ln |\mathbf{I} + \Sigma\Lambda_{\mathrm{MAP}}|. \tag{18}$$

We usually collect $\{\ln\kappa, \ln\sigma, b_1, \ln\Delta_2, \ldots, \ln\Delta_{r-1}\}$ as the set of variables to tune. This definition of tunable variables is helpful to convert the constrained optimization problem into an unconstrained optimization problem. The outline of our algorithm for model adaptation is described in Table 5.

Table 5: The outline of our algorithm for model adaptation using the MAP approach with Laplace approximation.

Initialization	choose a favorite gradient-descent optimization package select the starting point $\boldsymbol{\theta}$ for the optimization package
Looping	while the optimization package requests evidence/gradient evaluation at $\boldsymbol{\theta}$ 1. find the MAP estimate by solving the convex programming problem (9) 2. evaluate the negative logarithm of the evidence (18) at the MAP 3. calculate the gradients with respect to $\boldsymbol{\theta}$ (19)–(22) 4. feed the evidence and gradients to the optimization package
Exit	Return the optimal $\boldsymbol{\theta}$ found by the optimization package

The derivatives of $-\ln \mathcal{P}(\mathcal{D}|\boldsymbol{\theta})$ with respect to these variables can be derived as follows:

$$\begin{aligned}\frac{\partial - \ln \mathcal{P}(\mathcal{D}|\boldsymbol{\theta})}{\partial \ln \kappa} &= \frac{\kappa}{2}\mathrm{trace}\left[(\Lambda_{\mathrm{MAP}}^{-1}+\Sigma)^{-1}\frac{\partial \Sigma}{\partial \kappa}\right] - \frac{\kappa}{2}\boldsymbol{f}_{\mathrm{MAP}}^T\Sigma^{-1}\frac{\partial \Sigma}{\partial \kappa}\Sigma^{-1}\boldsymbol{f}_{\mathrm{MAP}} \\ &\quad + \frac{\kappa}{2}\mathrm{trace}\left[\Lambda_{\mathrm{MAP}}^{-1}(\Lambda_{\mathrm{MAP}}^{-1}+\Sigma)^{-1}\Sigma\frac{\partial \Lambda_{\mathrm{MAP}}}{\partial \kappa}\right];\end{aligned} \tag{19}$$

$$\frac{\partial - \ln \mathcal{P}(\mathcal{D}|\boldsymbol{\theta})}{\partial \ln \sigma} = \sigma\sum_{i=1}^{n}\frac{\partial \ell(y_i, f_{\mathrm{MAP}}(x_i))}{\partial \sigma} + \frac{\sigma}{2}\mathrm{trace}\left[\Lambda_{\mathrm{MAP}}^{-1}(\Lambda_{\mathrm{MAP}}^{-1}+\Sigma)^{-1}\Sigma\frac{\partial \Lambda_{\mathrm{MAP}}}{\partial \sigma}\right]; \tag{20}$$

$$\frac{\partial - \ln \mathcal{P}(\mathcal{D}|\boldsymbol{\theta})}{\partial b_1} = \sum_{i=1}^{n}\frac{\partial \ell(y_i, f_{\mathrm{MAP}}(x_i))}{\partial b_1} + \frac{1}{2}\mathrm{trace}\left[\Lambda_{\mathrm{MAP}}^{-1}(\Lambda_{\mathrm{MAP}}^{-1}+\Sigma)^{-1}\Sigma\frac{\partial \Lambda_{\mathrm{MAP}}}{\partial b_1}\right]; \tag{21}$$

$$\frac{\partial - \ln \mathcal{P}(\mathcal{D}|\boldsymbol{\theta})}{\partial \ln \Delta_\iota} = \Delta_\iota\sum_{i=1}^{n}\frac{\partial \ell(y_i, f_{\mathrm{MAP}}(x_i))}{\partial \Delta_\iota} + \frac{\Delta_\iota}{2}\mathrm{trace}\left[\Lambda_{\mathrm{MAP}}^{-1}(\Lambda_{\mathrm{MAP}}^{-1}+\Sigma)^{-1}\Sigma\frac{\partial \Lambda_{\mathrm{MAP}}}{\partial \Delta_\iota}\right]. \tag{22}$$

Note that at the MAP estimate $\Sigma^{-1}\boldsymbol{f}_{\mathrm{MAP}} = -\sum_{i=1}^{n}\left.\frac{\partial \ell(y_i, f(x_i))}{\partial \boldsymbol{f}}\right|_{\boldsymbol{f}=\boldsymbol{f}_{\mathrm{MAP}}}$. For more details, let us define

$$s_\rho = \frac{(z_1^i)^\rho \mathcal{N}(z_1^i;0,1)}{\Phi(z_1^i)-\Phi(z_2^i)}$$

and

$$v_\rho = \frac{(z_1^i)^\rho \mathcal{N}(z_1^i;0,1)-(z_2^i)^\rho \mathcal{N}(z_2^i;0,1)}{\Phi(z_1^i)-\Phi(z_2^i)}$$

where ρ runs from 0 to 3, $z_1^i = \frac{b_{y_i}-f(x_i)}{\sigma}$ and $z_2^i = \frac{b_{y_i-1}-f(x_i)}{\sigma}$. The ii-th entry of the diagonal matrix Λ is denoted as Λ_{ii}, which is defined as in (7), i.e. $\Lambda_{ii} = \frac{1}{\sigma^2}(v_0)^2 + \frac{1}{\sigma^2}v_1$. The detailed derivatives are given in the following:

- $\frac{\partial \Lambda_{ii}}{\partial \kappa} = \frac{\partial \Lambda_{ii}}{\partial \boldsymbol{f}^T}\frac{\partial \boldsymbol{f}}{\partial \kappa}$.
- $\frac{\partial \Lambda_{ii}}{\partial f(x_i)} = \frac{1}{\sigma^3}(2(v_0)^3 + 3v_0v_1 + v_2 - v_0)$.
- $\frac{\partial \boldsymbol{f}}{\partial \kappa} = \Lambda^{-1}(\Lambda^{-1}+\Sigma)^{-1}\frac{\partial \Sigma}{\partial \kappa}\Sigma^{-1}\boldsymbol{f}$.

- $\frac{\partial \ell(y_i, f(x_i))}{\partial \sigma} = \frac{v_1}{\sigma}$.
- $\frac{\partial \Lambda_{ii}}{\partial \sigma} = -\frac{2}{\sigma}\Lambda_{ii} + \frac{1}{\sigma^3}(2v_0v_2 + 2(v_0)^2v_1 - v_1 + (v_1)^2 + v_3) + \frac{\partial \Lambda_{ii}}{\partial \boldsymbol{f}^T}\frac{\partial \boldsymbol{f}}{\partial \sigma}$.
- $\frac{\partial \boldsymbol{f}}{\partial \sigma} = \Lambda^{-1}(\Lambda^{-1}+\Sigma)^{-1}\Sigma\boldsymbol{\psi}_\sigma$, where $\boldsymbol{\psi}_\sigma$ is a column vector whose i-th element is $\frac{1}{\sigma^2}(v_0 - v_0v_1 - v_2)$.
- $\frac{\partial \Lambda_{ii}}{\partial b_1} = -\frac{\partial \Lambda_{ii}}{\partial f(x_i)} + \frac{\partial \Lambda_{ii}}{\partial \boldsymbol{f}^T}\frac{\partial \boldsymbol{f}}{\partial b_1}$.
- $\frac{\partial \boldsymbol{f}}{\partial b_1} = \Lambda^{-1}(\Lambda^{-1} + \Sigma)^{-1}\Sigma\boldsymbol{\psi}_b$, where $\boldsymbol{\psi}_b$ is a column vector whose i-th element is Λ_{ii}.
- $\frac{\partial \ell(y_i, f(x_i))}{\partial \Delta_\iota} = \begin{cases} -\frac{v_0}{\sigma} & \text{if } \; y_i > \iota; \\ -\frac{s_0}{\sigma} & \text{if } \; y_i = \iota; \\ 0 & \text{otherwise.} \end{cases}$
- $\frac{\partial \Lambda_{ii}}{\partial \Delta_\iota} = \begin{cases} -\frac{\partial \Lambda_{ii}}{\partial f(x_i)} + \frac{\partial \Lambda_{ii}}{\partial \boldsymbol{f}^T}\frac{\partial \boldsymbol{f}}{\partial \Delta_\iota} & \text{if } \; y_i > \iota; \\ \varphi_i + \frac{\partial \Lambda_{ii}}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}^T}{\partial \Delta_\iota} & \text{if } \; y_i = \iota; \\ \frac{\partial \Lambda_{ii}}{\partial \boldsymbol{f}}\frac{\partial \boldsymbol{f}^T}{\partial \Delta_\iota} & \text{otherwise.} \end{cases}$
- $\varphi_i = \frac{\partial \Lambda_{ii}}{\partial \Delta_\iota} = \frac{1}{\sigma^3}(s_0 - 2v_0s_1 - 2(v_0)^2s_0 - s_2 - v_1s_0)$.
- $\frac{\partial \boldsymbol{f}}{\partial \Delta_\iota} = \Lambda^{-1}(\Lambda^{-1} + \Sigma)^{-1}\Sigma\boldsymbol{\psi}_\Delta$, where $\boldsymbol{\psi}_\Delta$ is a column vector whose i-th element is defined as $\boldsymbol{\psi}_\Delta^i = \begin{cases} \Lambda_{ii} \text{ i.e. } \frac{1}{\sigma^2}((v_0)^2 + v_1) & \text{if } \; y_i > \iota; \\ \frac{1}{\sigma^2}(v_0s_0 + s_1) & \text{if } \; y_i = \iota; \\ 0 & \text{otherwise.} \end{cases}$

Appendix B. Approximate posterior distribution by EP

The expectation propagation algorithm attempts to approximate $\mathcal{P}(\boldsymbol{f}|\mathcal{D})$ in form of a product of Gaussian distributions $Q(\boldsymbol{f}) = \prod_{i=1}^n \tilde{t}(f(x_i))\mathcal{P}(\boldsymbol{f})$ where $\tilde{t}(f(x_i)) = s_i \exp(-\frac{1}{2}p_i(f(x_i) - m_i)^2)$. The updating scheme is given as follows.

The initial states:

- individual mean $m_i = 0 \; \forall i$;
- individual inverse variance $p_i = 0 \; \forall i$;
- individual amplitude $s_i = 1 \; \forall i$;
- posterior covariance $\mathcal{A} = (\Sigma^{-1} + \Pi)^{-1}$, where $\Pi = \mathrm{diag}(p_1, p_2, \ldots, p_n)$;
- posterior mean $\boldsymbol{h} = \mathcal{A}\Pi\boldsymbol{m}$, where $\boldsymbol{m} = [m_1, m_2, \ldots, m_n]^T$.

Looping i from 1 to n until there is no significant change in $\{m_i, p_i, s_i\}_{i=1}^n$:

- $\tilde{t}(f(x_i))$ is removed from $Q(\boldsymbol{f})$ to get a leave-one-out posterior distribution $Q^{\setminus i}(\boldsymbol{f})$ having

- variance of $f(x_i)$: $\lambda_i^{\backslash i} = \frac{\mathcal{A}_{ii}}{1-\mathcal{A}_{ii}p_i}$;
- mean of $f(x_i)$: $h_i^{\backslash i} = h_i + \lambda_i^{\backslash i} p_i(h_i - m_i)$;
- others with $j \neq i$: $\lambda_j^{\backslash i} = \mathcal{A}_{jj}$ and $h_j^{\backslash i} = h_j$.

- $\tilde{t}(f(x_i))$ in $Q(\boldsymbol{f})$ is updated by incorporating the message $\mathcal{P}(y_i|f(x_i))$ into $Q^{\backslash i}(\boldsymbol{f})$:
  - $\mathcal{Z}_i = \int \mathcal{P}(y_i|f(x_i))\mathcal{N}(f(x_i); h_i^{\backslash i}, \lambda_i^{\backslash i})df(x_i) = \Phi(\tilde{z}_1) - \Phi(\tilde{z}_2)$
    where $\tilde{z}_1 = \frac{b_{y_i} - h_i^{\backslash i}}{\sqrt{\lambda_i^{\backslash i} + \sigma^2}}$ and $\tilde{z}_2 = \frac{b_{y_i - 1} - h_i^{\backslash i}}{\sqrt{\lambda_i^{\backslash i} + \sigma^2}}$.
  - $$\beta_i = \frac{\partial \log \mathcal{Z}_i}{\partial \lambda_i^{\backslash i}} = -\frac{1}{2(\lambda_i^{\backslash i} + \sigma^2)} \left(\frac{\tilde{z}_1 \mathcal{N}(\tilde{z}_1;0,1) - \tilde{z}_2 \mathcal{N}(\tilde{z}_2;0,1)}{\Phi(\tilde{z}_1) - \Phi(\tilde{z}_2)} \right).$$
    $$\gamma_i = \frac{\partial \log \mathcal{Z}_i}{\partial h_i^{\backslash i}} = -\frac{1}{\sqrt{\lambda_i^{\backslash i} + \sigma^2}} \left(\frac{\mathcal{N}(\tilde{z}_1;0,1) - \mathcal{N}(\tilde{z}_2;0,1)}{\Phi(\tilde{z}_1) - \Phi(\tilde{z}_2)} \right). \tag{23}$$
  - $\upsilon_i = \gamma_i^2 - 2\beta_i$.
  - $h_i^{new} = h_i^{\backslash i} + \lambda_i^{\backslash i}\gamma_i$.
  - $p_i^{new} = \frac{\upsilon_i}{1 - \lambda_i^{\backslash i}\upsilon_i}$.
  - $m_i^{new} = h_i^{\backslash i} + \frac{\gamma_i}{\upsilon_i}$.
  - $s_i^{new} = \mathcal{Z}_i \sqrt{\lambda_i^{\backslash i} p_i^{new} + 1} \exp\left(\frac{\gamma_i^2}{2\upsilon_i}\right)$.

- Note that $p_i^{new} > 0$ all the time, because $0 < \upsilon_i < \frac{1}{\lambda_i^{\backslash i} + \sigma^2}$ and then $\lambda_i^{\backslash i}\upsilon_i < 1$.

- if $p_i^{new} \approx p_i$, skip this sample and this updating; otherwise update $\{p_i, m_i, s_i\}$, the posterior mean $\boldsymbol{h}$ and covariance $\mathcal{A}$ as follows:
  - $\mathcal{A}^{new} = \mathcal{A} - \rho \boldsymbol{a}_i \boldsymbol{a}_i^T$ where $\rho = \frac{p_i^{new} - p_i}{1 + (p_i^{new} - p_i)\mathcal{A}_{ii}}$ and $\boldsymbol{a}_i$ is the i-th column of $\mathcal{A}$.
  - $\boldsymbol{h}^{new} = h + \eta \boldsymbol{a}_i$ where $\eta = \frac{\gamma_i + p_i(h_i - m_i)}{1 - \mathcal{A}_{ii}p_i}$ and γ_i is defined as in (23).

As a byproduct, we can get the approximate evidence $\mathcal{P}(\mathcal{D}|\boldsymbol{\theta})$ at the EP solution, which can be written as
$$\prod_{i=1}^{n} s_i \frac{\det^{\frac{1}{2}}(\Pi^{-1})}{\det^{\frac{1}{2}}(\Sigma + \Pi^{-1})} \exp\left(\frac{B}{2}\right)$$
where $B = \sum_{ij} \mathcal{A}_{ij}(m_i p_i)(m_j p_j) - \sum_i p_i m_i^2$.

Appendix C. Gradient formulae for variational bound

At the equilibrium of $Q(\boldsymbol{f})$, the variational bound $\mathcal{F}(\boldsymbol{\theta})$ can be analytically calculated as follows:
$$\begin{aligned}\mathcal{F}(\boldsymbol{\theta}) = & \sum_{i=1}^{n} \int \mathcal{N}(f(x_i); h_i, \mathcal{A}_{ii}) \ln(\mathcal{P}(y_i|f(x_i)))df(x_i) - \frac{1}{2}\ln|\mathrm{I} + \Sigma\Pi| \\ & -\frac{1}{2}\mathrm{trace}((\mathrm{I} + \Sigma\Pi)^{-1}) - \frac{1}{2}\boldsymbol{m}^T(\Sigma + \Pi^{-1})^{-1}\Sigma(\Sigma + \Pi^{-1})^{-1}\boldsymbol{m} + \frac{n}{2} \ .\end{aligned}$$

Note that $(\Sigma+\Pi^{-1})^{-1}\boldsymbol{m}$ can be directly obtained by $\{\gamma_i\}$ defined as in (23). The gradient of $\mathcal{F}(\boldsymbol{\theta})$ with respect to the variables $\{\ln\kappa, \ln\sigma, b_1, \ln\Delta_2, \ldots, \ln\Delta_{r-1}\}$ can be given in the following:

$$\begin{aligned}
\frac{\partial\mathcal{F}(\boldsymbol{\theta})}{\partial\ln\kappa} &= \kappa\int Q(\boldsymbol{f})\frac{\partial\log\mathcal{P}(\boldsymbol{f})}{\partial\kappa}d\boldsymbol{f}\\
&= -\frac{\kappa}{2}\mathrm{trace}\left(\Sigma^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\kappa}}\right)+\frac{\kappa}{2}\boldsymbol{h}^T\Sigma^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\kappa}}\Sigma^{-1}\boldsymbol{h}+\frac{\kappa}{2}\mathrm{trace}\left(\Sigma^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\kappa}}\Sigma^{-1}\mathcal{A}\right)\\
&= -\frac{\kappa}{2}\mathrm{trace}\left((\Pi^{-1}+\Sigma)^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\kappa}}\right)+\frac{\kappa}{2}\boldsymbol{m}^T(\Pi^{-1}+\Sigma)^{-1}\frac{\partial\Sigma}{\partial\boldsymbol{\kappa}}(\Pi^{-1}+\Sigma)^{-1}\boldsymbol{m}\ ,
\end{aligned}$$

$$\begin{aligned}
\frac{\partial\mathcal{F}(\boldsymbol{\theta})}{\partial\ln\sigma} &= \sigma\textstyle\sum_{i=1}^n\int\mathcal{N}(f(x_i);h_i,\mathcal{A}_{ii})\frac{\partial\ln\mathcal{P}(y_i|f(x_i))}{\partial\sigma}df(x_i)\\
&= -\textstyle\sum_{\{1\le y_i<r\}}\int\mathcal{N}\left(f(x_i);\frac{h_i\sigma^2+\mathcal{A}_{ii}b_{y_i}}{\sigma^2+\mathcal{A}_{ii}},\frac{\sigma^2\mathcal{A}_{ii}}{\sigma^2+\mathcal{A}_{ii}}\right)\frac{\frac{b_{y_i}-f(x_i)}{\sqrt{2\pi(\sigma^2+\mathcal{A}_{ii})}}\exp\left(-\frac{(h_i-b_{y_i})^2}{2(\sigma^2+\mathcal{A}_{ii})}\right)}{\mathcal{P}(y_i|f(x_i))}df(x_i)\\
&\quad+\textstyle\sum_{\{1<y_i\le r\}}\int\mathcal{N}\left(f(x_i);\frac{h_i\sigma^2+\mathcal{A}_{ii}b_{y_i-1}}{\sigma^2+\mathcal{A}_{ii}},\frac{\sigma^2\mathcal{A}_{ii}}{\sigma^2+\mathcal{A}_{ii}}\right)\frac{\frac{b_{y_i-1}-f(x_i)}{\sqrt{2\pi(\sigma^2+\mathcal{A}_{ii})}}\exp\left(-\frac{(h_i-b_{y_i-1})^2}{2(\sigma^2+\mathcal{A}_{ii})}\right)}{\mathcal{P}(y_i|f(x_i))}df(x_i)\ ,\\
\frac{\partial\mathcal{F}(\boldsymbol{\theta})}{\partial b_1} &= \textstyle\sum_{i=1}^n\int\mathcal{N}(f(x_i);h_i,\mathcal{A}_{ii})\frac{\partial\ln\mathcal{P}(y_i|f(x_i))}{\partial b_1}df(x_i)\\
&= \textstyle\sum_{\{1\le y_i<r\}}\int\mathcal{N}(f(x_i);\frac{h_i\sigma^2+\mathcal{A}_{ii}b_{y_i}}{\sigma^2+\mathcal{A}_{ii}},\frac{\sigma^2\mathcal{A}_{ii}}{\sigma^2+\mathcal{A}_{ii}})\frac{\frac{1}{\sqrt{2\pi(\sigma^2+\mathcal{A}_{ii})}}\exp\left(-\frac{(h_i-b_{y_i})^2}{2(\sigma^2+\mathcal{A}_{ii})}\right)}{\mathcal{P}(y_i|f(x_i)}df(x_i)\\
&\quad-\textstyle\sum_{\{1<y_i\le r\}}\int\mathcal{N}(f(x_i);\frac{h_i\sigma^2+\mathcal{A}_{ii}b_{y_i-1}}{\sigma^2+\mathcal{A}_{ii}},\frac{\sigma^2\mathcal{A}_{ii}}{\sigma^2+\mathcal{A}_{ii}})\frac{\frac{1}{\sqrt{2\pi(\sigma^2+\mathcal{A}_{ii})}}\exp\left(-\frac{(h_i-b_{y_i-1})^2}{2(\sigma^2+\mathcal{A}_{ii})}\right)}{\mathcal{P}(y_i|f(x_i))}df(x_i)\ ,\\
\frac{\partial\mathcal{F}(\boldsymbol{\theta})}{\partial\ln\Delta_\iota} &= \Delta_\iota\textstyle\sum_{i=1}^n\int\mathcal{N}(f(x_i);h_i,\mathcal{A}_{ii})\frac{\partial\ln\mathcal{P}(y_i|f(x_i))}{\partial\Delta_\iota}df(x_i)\\
&= \Delta_\iota\textstyle\sum_{\{\iota\le y_i<r\}}\int\mathcal{N}(f(x_i);\frac{h_i\sigma^2+\mathcal{A}_{ii}b_{y_i}}{\sigma^2+\mathcal{A}_{ii}},\frac{\sigma^2\mathcal{A}_{ii}}{\sigma^2+\mathcal{A}_{ii}})\frac{\frac{1}{\sqrt{2\pi(\sigma^2+\mathcal{A}_{ii})}}\exp\left(-\frac{(h_i-b_{y_i})^2}{2(\sigma^2+\mathcal{A}_{ii})}\right)}{\mathcal{P}(y_i|f(x_i)}df(x_i)\\
&\quad-\Delta_\iota\textstyle\sum_{\{\iota<y_i\le r\}}\int\mathcal{N}(f(x_i);\frac{h_i\sigma^2+\mathcal{A}_{ii}b_{y_i-1}}{\sigma^2+\mathcal{A}_{ii}},\frac{\sigma^2\mathcal{A}_{ii}}{\sigma^2+\mathcal{A}_{ii}})\frac{\frac{1}{\sqrt{2\pi(\sigma^2+\mathcal{A}_{ii})}}\exp\left(-\frac{(h_i-b_{y_i-1})^2}{2(\sigma^2+\mathcal{A}_{ii})}\right)}{\mathcal{P}(y_i|f(x_i))}df(x_i)\ ,
\end{aligned}$$

where $\sum_{\{\iota<y_i\le r\}}$ means summing over all the samples whose targets satisfy $\iota<y_i\le r$, and these one-dimensional integrals can be approximated using Gaussian quadrature or calculated by Romberg integration at some appropriate accuracy.

References

Justin Basilico and Thomas Hofmann. Unifying collaborative and content-based filtering. In *Proceedings of the 21th International Conference on Machine Learning*, pages 65–72, 2004.

H. J. Brascamp and E. H. Lieb. On extensions of the Brunn-Minkowski and Prekopa-Leindler theorems, including inequalities for log concave functions, and with an application to the diffusion equation. *Journal of Functional Analysis*, 22:366–389, 1976.

R. H. Byrd, P. Lu, and J. Nocedal. A limited memory algorithm for bound constrained optimization. *SIAM Journal on Scientific and Statistical Computing*, 16(5):1190–1208, 1995.

W. Chu and S. S. Keerthi. New approaches to support vector ordinal regression. Technical report, Yahoo! Research Labs, 2005.

William W. Cohen, Robert E. Schapire, and Yoram Singer. Learning to order things. *Journal of artificial intelligence research*, 10:243–270, 1999.

Compaq. EachMovie. *http://research.compaq.com/SRC/eachmovie/*, 2001.

K. Crammer and Y. Singer. Pranking with ranking. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 641–647, Cambridge, MA, 2002. MIT Press.

L. Csató, E. Fokoué, M. Opper, B. Schottky, and O. Winther. Efficient approaches to Gaussian process classification. In Sara A. Solla, Todd K. Leen, and Klaus-Robert Müller, editors, *Advances in Neural Information Processing Systems 12*, pages 251–257, 2000.

L. Csató and M. Opper. Sparse online Gaussian processes. *Neural Computation, The MIT Press*, 14:641–668, 2002.

L. Fahrmeir and G. Tutz. *Multivariate Statistical Modelling Based on Generalized Linear Models*. New York, Springer-Verlag, 2nd edition, 2001.

Eibe Frank and Mark Hall. A simple approach to ordinal classification. In *Proceedings of the European Conference on Machine Learning*, pages 145–165, 2001.

S. Har-Peled, D. Roth, and D. Zimak. Constraint classification: A new approach to multiclass classification and ranking. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 785–792, 2003.

T. Hastie and R. Tibshirani. *Generalized Additive Models*. Chapman and Hall, London, 1990.

R. Herbrich, T. Graepel, and K. Obermayer. Large margin rank boundaries for ordinal regression. In *Advances in Large Margin Classifiers*, pages 115–132. MIT Press, 2000.

Valen E. Johnson and James H. Albert. *Ordinal Data Modeling (Statistics for Social Science and Public Policy)*. Springer-Verlag, 1999.

S. S. Keerthi, S. K. Shevade, C. Bhattacharyya, and K. R. K. Murthy. Improvements to Platt's SMO algorithm for SVM classifier design. *Neural Computation*, 13:637–649, March 2001.

H. Kim and Z. Ghahramani. The EM-EP algorithm for Gaussian process classification. In *Proc. of the Workshop on Probabilistic Graphical Models for Classification (at ECML)*, 2003.

S. Kramer, G. Widmer, B. Pfahringer, and M. DeGroeve. Prediction of ordinal classes using regression trees. *Fundamenta Informaticae*, 47:1–13, 2001.

Gert R. G. Lanckriet, Nello Cristianini, Peter Bartlett, Laurent El Ghaoui, and Michael I. Jordan. Learning the kernel matrix with semidefinite programming. *Journal of Machine Learning Research*, 5:27–72, 2004.

Neil D. Lawrence, Matthias Seeger, and Ralf Herbrich. Fast sparse Gaussian process methods: the informative vector machine. In S. Becker, S. Thrun, and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 609–616, 2003.

D. J. C. MacKay. A practical Bayesian framework for back propagation networks. *Neural Computation*, 4(3):448–472, 1992.

D. J. C. MacKay. Bayesian methods for backpropagation networks. *Models of Neural Networks III*, pages 211–254, 1994.

P. McCullagh. Regression models for ordinal data. *Journal of the Royal Statistical Society B*, 42(2):109–142, 1980.

P. McCullagh and J. A. Nelder. *Generalized Linear Models.* Chapman & Hall, London, 1983.

T. P. Minka. *A family of algorithms for approximate Bayesian inference.* PhD thesis, Massachusetts Institute of Technology, January 2001.

R. M. Neal. *Bayesian Learning for Neural Networks.* Lecture Notes in Statistics. Springer, 1996.

R. M. Neal. Monte Carlo implementation of Gaussian process models for Bayesian regression and classification. Technical Report No. 9702, Department of Statistics, University of Toronto, 1997.

A. O'Hagan. Curve fitting and optimal design for prediction (with discussion). *Journal of the Royal Statistical Society B*, 40(1):1–42, 1978.

John W. Pratt. Concavity of the log likelihood. *Journal of the American Statistical Association*, 76(373):103–106, 1981.

Carl Edward Rasmussen and Zoubin Ghahramani. Infinite mixtures of Gaussian process experts. In T. G. Dietterich, S. Becker, and Z. Ghahramani, editors, *Advances in Neural Information Processing Systems 14*, pages 881–888, 2002.

B. Schölkopf and A. J. Smola. *Learning with Kernels – Support Vector Machines, Regularization, Optimization and Beyond.* Adaptive Computation and Machine Learning. The MIT Press, December 2001.

M. Seeger. Notes on Minka's expectation propagation for Gaussian process classification. Technical report, University of Edinburgh, 2002.

M. Seeger. *Bayesian Gaussian process models: PAC-Bayesian generalisation error bounds and sparse approximations.* PhD thesis, University of Edinburgh, July 2003.

Amnon Shashua and Anat Levin. Ranking with large margin principle: two approaches. In S. Thrun S. Becker and K. Obermayer, editors, *Advances in Neural Information Processing Systems 15*, pages 937–944. MIT Press, 2003.

D. Singh, P. G. Febbo, K. Ross, D. G. Jackson, J. Manola, C. Ladd, P. Tamayo, A. A. Renshaw, A. V. D'Amico, J. P. Richie, E. S. Lander, M. Loda, P. W. Kantoff, T. R. Golub, and W. R. Sellers. Gene expression correlates of clinical prostate cancer behavior. *Cancer Cell*, 1:203–209, 2002. www.genome.wi.mit.edu/MPR/prostate.

E. Snelson, Zoubin Ghahramani, and Carl Rasmussen. Warped Gaussian processes. In Sebastian Thrun, Lawrence Saul, and Bernhard Schölkopf, editors, *Advances in Neural Information Processing Systems 16*, 2004.

Nathan Srebro and Tommi Jaakkola. Weighted low-rank approximations. In *Proceedings of the Twentieth International Conference on Machine Learning*, 2003.

Volker Tresp. A Bayesian committee machine. *Neural Computation*, 12(11):2719–2741, November 2000.

Gerhard Tutz. Generalized semiparametrically structured ordinal models. *Biometrics*, 59: 263–273, June 2003.

V. N. Vapnik. *The Nature of Statistical Learning Theory.* New York: Springer-Verlag, 1995.

G. Wahba. *Spline Models for Observational Data*, volume 59 of *CBMS-NSF Regional Conference Series in Applied Mathematics.* SIAM, 1990.

C. K. I. Williams and D. Barber. Bayesian classification with Gaussian processes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(12):1342–1351, 1998.

C. K. I. Williams and C. E. Rasmussen. Gaussian processes for regression. In D. S. Touretzky, M. C. Mozer, and M. E. Hasselmo, editors, *Advances in Neural Information Processing Systems*, volume 8, pages 598–604, 1996. MIT Press.