A Probabilistic Least Squares Approach to Ordinal Regression

P.K. Srijith[1], Shirish Shevade[1], and S. Sundararajan[2]

[1] Computer Science and Automation, Indian Institute of Science, India
{srijith,shirish}@csa.iisc.ernet.in
[2] Yahoo! Labs, Bangalore, India
zensid@yahoo.com

Abstract. This paper proposes a novel approach to solve the ordinal regression problem using Gaussian processes. The proposed approach, probabilistic least squares ordinal regression (PLSOR), obtains the probability distribution over ordinal labels using a particular likelihood function. It performs model selection (hyperparameter optimization) using the leave-one-out cross-validation (LOO-CV) technique. PLSOR has conceptual simplicity and ease of implementation of least squares approach. Unlike the existing Gaussian process ordinal regression (GPOR) approaches, PLSOR does not use any approximation techniques for inference. We compare the proposed approach with the state-of-the-art GPOR approaches on some synthetic and benchmark data sets. Experimental results show the competitiveness of the proposed approach.

Keywords: Gaussian processes, ordinal regression, probabilistic least squares, cross-validation.

1 Introduction

Most of the works in machine learning have focused on the standard problems of classification and regression. Classification problems aim to label examples from a discrete unordered set, while regression problems aim to label examples from a real valued set. Recently some new classes of learning problems started emerging and the prominent among them is the ordinal regression problem [1]. This problem aims to provide labels to the examples from a discrete but ordered set. It differs from a multi-class classification problem in that the labels are ordered, and from a regression problem in that the labels are discrete. The problem arises in social sciences and information retrieval, where humans rate an item on an ordinal scale. In information retrieval, a user may grade the retrieved documents as highly relevant, relevant, irrelevant or highly irrelevant.

Formally, we define the ordinal regression problem as follows. We are given a sample of n labeled independent training examples, $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{n}$, where x_i is an element of a d dimensional input space X ($X \subseteq \mathcal{R}^d$) and y_i is an element of output space Y. The output space $Y = \{r_1, r_2, \ldots, r_q\}$ is a discrete set with an order among its elements, say $r_1 < r_2 < \ldots < r_q$. Our goal is to learn a decision

M. Thielscher and D. Zhang (Eds.): AI 2012, LNCS 7691, pp. 683–694, 2012.

function $h : X \to Y$, such that it generalizes well. We consider an ordinal regression problem with r ordered categories and without loss of generality, we denote them by r consecutive integers $\{1, 2, \ldots, r\}$. Ordinal regression problems have the property that the penalty for an incorrect prediction should be proportional to the deviation of the predicted label from the true label.

Most of the works on ordinal regression problems are based on the large margin framework [1,2,3]. A distribution independent learning approach based on a loss function between pairs of examples was used in [1] to perform ordinal regression. Fixed margin and sums of margin approaches [2] used the support vector machine framework to solve the ordinal regression problem. They learn $r - 1$ thresholds that divide the real line into r consecutive intervals for r ordered categories. However the thresholds learnt with this approach need not be ordered. Support vector ordinal regression [3] approach corrected this problem by explicitly specifying the ordering constraint on the thresholds. It also proposed a new formulation which implicitly takes into account the ordering constraint on the thresholds. Kernel discriminant ordinal regression [4] extended the Kernel discriminant learning for classification to the ordinal regression setting. In sparse Bayesian ordinal regression [5], the proportional odds model [6] for ordinal regression is extended using kernel methods, and a sparse solution is obtained by imposing a zero-mean Gaussian prior distribution over the weight vector.

Gaussian processes (GP) are non parametric Bayesian models which provide a probabilistic approach to learning in a kernel based framework [7]. The existing Gaussian process approaches for ordinal regression [8] use a non Gaussian likelihood function for modeling the ordinal labels. The use of non Gaussian likelihood forces it to use approximation methods like Laplace approximation [7] or expectation propagation [9] to obtain an approximate Gaussian posterior. The approach performs model selection by maximizing the marginal likelihood using either a maximum a posteriori approach (MAP-GPOR) or expectation propagation approach (EP-GPOR). MAP-GPOR and EP-GPOR are among the state-of-the-art approaches for ordinal regression.

In this work, we propose a simple approach, probabilistic least squares ordinal regression (PLSOR), to perform ordinal regression using Gaussian processes. In PLSOR, the predictive distribution of the latent functions is learnt as a Gaussian process regression (GPR) on ordinal variables. This results in a Gaussian distributed posterior which avoids the use of any approximation methods. The predictive distribution of ordinal targets is obtained by using a likelihood function which takes care of the regression nature of the latent function. In PLSOR, the model parameters are estimated using leave-one-out cross-validation (LOO-CV) [7]. The experiments on synthetic and benchmark data sets showed that the performance of the PLSOR approach is comparable with that of the MAP-GPOR and EP-GPOR approaches. This is also validated using a statistical significance test.

The rest of the paper is organized as follows. In Section 2, we discuss Gaussian process regression. The MAP-GPOR and EP-GPOR approaches are summarized in Section 3. Section 4 discusses the proposed approach, probabilistic

least squares ordinal regression (PLSOR), in detail. Comparison of PLSOR with the MAP-GPOR and EP-GPOR approaches on synthetic and benchmark data sets is presented in Section 5. Finally, some conclusions are drawn in Section 6.

2 Gaussian Process Regression

A Gaussian process (GP) is a collection of random variables with the property that the joint distribution of any finite subset of which is a Gaussian [7]. It generalizes Gaussian distribution to infinitely many random variables. The GP is completely specified by a mean function and a covariance function. The covariance function is defined over function values of a pair of input and is evaluated using the Mercer kernel function over the pair of inputs. The covariance function expresses some general properties of functions such as their smoothness, and length-scale. A commonly used covariance function is squared exponential (SE) or Gaussian kernel

$$cov\big(f(x_i), f(x_j)\big) = K(x_i, x_j) = \sigma_f^2 \exp(-\frac{\kappa}{2}||x_i - x_j||^2). \tag{1}$$

Here $f(x_i)$ and $f(x_j)$ are function values associated with the inputs x_i and x_j respectively. σ_f^2 and $\kappa > 0$ are hyperparameters associated with the covariance function.

In a regression problem the output space Y is real valued, *i.e.* $Y \subseteq \mathcal{R}$. We assume a noisy Gaussian process regression (GPR) approach in which the outputs lie around a latent function $f(x)$ with an additive, independently and identically distributed (i.i.d.) Gaussian noise ϵ with mean 0 and variance σ_n^2, *i.e.* $y = f(x) + \epsilon$. The likelihood function for the noisy GPR approach follows a Gaussian distribution

$$p(y|f(x)) = \mathcal{N}(f(x), \sigma_n^2). \tag{2}$$

Let $\mathcal{D}$ be the set consisting of n training data points $\mathbf{X}$, the corresponding outputs $\mathbf{y}$ and n_* test data points $\mathbf{X}_*$. Let $\mathbf{K} = K(\mathbf{X}, \mathbf{X})$, $\mathbf{K}_* = K(\mathbf{X}, \mathbf{X}_*)$ and $\mathbf{K}_{**} = K(\mathbf{X}_*, \mathbf{X}_*)$. Here $K(\mathbf{X}, \mathbf{X}_*)$ is an $n \times n_*$ matrix of covariances evaluated at all pairs of training and test input data. The matrices $K(\mathbf{X}, \mathbf{X})$, $K(\mathbf{X}_*, \mathbf{X})$ and $K(\mathbf{X}_*, \mathbf{X}_*)$ are also defined similarly. The GPR approach imposes a zero mean GP prior over the training latent functions $\mathbf{f}$ and test latent functions $\mathbf{f}_*$. The predictive distribution for the test latent functions, $p(\mathbf{f}_*|\mathcal{D})$, is obtained by integrating the conditional distribution $p(\mathbf{f}_*|\mathbf{f}, \mathcal{D})$ over the posterior distribution $p(\mathbf{f}|\mathcal{D})$, *i.e.* $p(\mathbf{f}_*|\mathcal{D}) = \int p(\mathbf{f}_*|\mathbf{f}, \mathcal{D}) p(\mathbf{f}|\mathcal{D}) d\mathbf{f}$. In GPR, both the conditional distribution and the posterior distribution are multivariate Gaussians. Hence the predictive distribution of the test latent functions is a multivariate Gaussian with mean (4) and covariance (5),

$$p(\mathbf{f}_*|\mathcal{D}) = \mathcal{N}(\bar{\mathbf{f}}_*, \mathrm{cov}(\mathbf{f}_*)), \text{where} \tag{3}$$

$$\bar{\mathbf{f}}_* = \mathbf{K}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{y}, \tag{4}$$

$$\mathrm{cov}(\mathbf{f}_*) = \mathbf{K}_{**} - \mathbf{K}_*^\top (\mathbf{K} + \sigma_n^2 \mathbf{I})^{-1} \mathbf{K}_*. \tag{5}$$

The predictive distribution of the test outputs $\mathbf{y}_*$ is obtained by averaging the likelihood (2) over predictive distribution of $\mathbf{f}_*$, $p(\mathbf{y}_*|\mathcal{D}) = \int p(\mathbf{y}_*|\mathbf{f}_*)p(\mathbf{f}_*|\mathcal{D})d\mathbf{f}_*$. It is also a multivariate Gaussian with the mean same as that of $\mathbf{f}_*$ while the covariance is obtained by adding $\sigma_n^2\mathbf{I}$ to the variance of $\mathbf{f}_*$. Model selection (hyperparameter optimization) is done using either Bayesian techniques or cross-validation techniques [7].

Performing ordinal regression using GPR is simple and straightforward. It treats the ordinal outputs as real numbers and perform regression on the ordinal outputs. However, such an approach does not provide a valid probability distribution over the ordinal outputs. The Gaussian process ordinal regression (GPOR) approaches [8], maximum a posteriori GPOR (MAP-GPOR) and expectation propagation GPOR (EP-GPOR), provide a valid probability distribution over the ordinal outputs. The following section briefly summarizes the MAP-GPOR and EP-GPOR approaches.

3 Gaussian Process Ordinal Regression Approaches

MAP-GPOR and EP-GPOR use a zero mean Gaussian process prior. Under noisy observations, for an input x and the latent function f, the likelihood function for an ordinal output y is defined as [8]

$$p(y|f) = \Phi\left(\frac{b_y - f}{\sigma}\right) - \Phi\left(\frac{b_{y-1} - f}{\sigma}\right) \tag{6}$$

where σ is the standard deviation of the Gaussian noise and Φ is the Gaussian cumulative distribution function *i.e.* $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(\delta; 0, 1)d\delta$. The thresholds $b_0, b_1, \ldots, b_r \in \mathcal{R}$ ($b_0 \leq b_1 \leq \ldots \leq b_r$ where $b_0 = -\infty$ and $b_r = \infty$) are fixed so that the likelihood function represents a valid probability distribution over the ordinal outputs. The thresholds $b_1 \leq b_2 \leq \ldots \leq b_{r-1}$ divide a real line into r contiguous intervals. A real latent function value is mapped to a discrete ordinal output based on the interval in which it lies. The likelihood (6) is not a Gaussian and therefore the posterior, $p(\mathbf{f}|\mathcal{D})$, is also not a Gaussian. MAP-GPOR works by approximating the posterior as a Gaussian distribution using Laplace approximation while EP-GPOR uses expectation propagation (EP) [9]. The MAP-GPOR and EP-GPOR approaches perform model selection by maximizing the evidence $p(\mathcal{D}|\theta)$, where θ is the model parameter vector which includes the kernel parameter κ in the covariance function[1], the threshold parameters$(b_1, b_2, \ldots, b_{r-1})$ and the noise parameter σ in the likelihood function. In MAP-GPOR, model selection is done using maximum a posteriori approach with Laplace approximation while in EP-GPOR, it is done using expectation propagation approach with variational methods. Both MAP-GPOR and EP-GPOR take $\mathcal{O}(n^3)$ time for model selection as the optimization method requires inversion of an $n \times n$ matrix.

[1] GPOR approach uses a squared exponential covariance function with a single hyperparameter κ. $cov(f(x_i), f(x_j)) = K(x_i, x_j) = \exp(-\frac{\kappa}{2}||x_i - x_j||^2)$, where $f(x_i)$ and $f(x_j)$ are function values associated with the inputs x_i and x_j respectively.

Performing ordinal regression using MAP-GPOR and EP-GPOR is complicated since they use a non Gaussian likelihood. They have to use approximation methods like Laplace approximation or expectation propagation to obtain a Gaussian posterior. We propose a new approach, probabilistic least squares ordinal regression (PLSOR), which provides a simple and exact way to perform ordinal regression using Gaussian processes.

4 Probabilistic Least Squares Ordinal Regression

PLSOR extends probabilistic least squares approach for classification [7] to the ordinal regression setting. In PLSOR, the predictive distribution of test latent functions is learnt using Gaussian process regression on ordinal outputs. The predictive distribution of the test outputs is learnt by squashing a linear function of test latent function predictive probability through a sigmoid. Since the test latent function f_* is learnt using GPR on ordinal outputs it takes real values ranging from 1 to r (number of ordinal categories). We map f_* to a real line by using a linear map $(\hat{\alpha} f_* + \hat{\beta})$, where $\hat{\alpha}, \hat{\beta} \in \mathcal{R}$. The real line is divided into r contiguous segments using thresholds $b_1 \leq b_2 \leq \ldots \leq b_{r-1}$. The segment (b_{y_*-1}, b_{y_*}) is associated with the ordinal category y_* and maps the scaled latent function value to that category. In PLSOR, the following likelihood function is used to estimate the probability of an ordinal category y_* for the test data x_*:

$$p(y_*|f_*) = \Phi(b_{y_*} - (\hat{\alpha} f_* + \hat{\beta})) - \Phi(b_{y_*-1} - (\hat{\alpha} f_* + \hat{\beta})). \tag{7}$$

Here $y_* \in \{1, 2, \ldots, r\}$, $b_0, \ldots, b_r \in \mathcal{R}$ such that $b_0 \leq b_1 \ldots \leq b_r$ and Φ denotes the Gaussian cumulative distribution function *i.e.* $\Phi(z) = \int_{-\infty}^{z} \mathcal{N}(\delta; 0, 1) d\delta$. We fix $b_0 = -\infty$ and $b_r = \infty$, so that the likelihood function is a valid probability distribution. The predictive distribution of the test latent function f_* is Gaussian with mean μ_* and variance σ_*^2 given by (4) and (5) respectively. The predictive distribution of the test ordinal category y_* is obtained by averaging the likelihood (7) over the test latent function predictive distribution:

$$\begin{aligned} p(y_*|x_*, \mathbf{X}, \mathbf{y}, \theta) = & \int \Phi(b_{y_*} - (\hat{\alpha} f_* + \hat{\beta})) \mathcal{N}(f_*|\mu_*, \sigma_*^2) df_* \\ & - \int \Phi(b_{y_*-1} - (\hat{\alpha} f_* + \hat{\beta})) \mathcal{N}(f_*|\mu_*, \sigma_*^2) df_* \\ = & \Phi\left(\frac{b_{y_*} - (\hat{\alpha}\mu_* + \hat{\beta})}{\sqrt{1 + \hat{\alpha}^2 \sigma_*^2}} \right) - \Phi\left(\frac{b_{y_*-1} - (\hat{\alpha}\mu_* + \hat{\beta})}{\sqrt{1 + \hat{\alpha}^2 \sigma_*^2}} \right). \end{aligned} \tag{8}$$

The predictive distribution (8) is redefined as

$$p(y_*|x_*, \mathbf{X}, \mathbf{y}, \theta) = \Phi\left(\frac{\alpha\mu_* + \beta_{y_*}}{\sqrt{1 + \alpha^2 \sigma_*^2}} \right) - \Phi\left(\frac{\alpha\mu_* + \beta_{y_*-1}}{\sqrt{1 + \alpha^2 \sigma_*^2}} \right) \tag{9}$$

where $\alpha \in \mathcal{R}$, $\beta_0 = -\infty$, $\beta_r = \infty$, $\beta_1, \beta_2, \ldots, \beta_{r-1} \in \mathcal{R}$ such that $\beta_1 \leq \beta_2 \leq \ldots \leq \beta_{r-1}$. Here we have redefined the variables as $\alpha = -\hat{\alpha}$ and $\beta_i = b_i - \hat{\beta}$. θ

is a vector of model parameters which include α, thresholds $(\beta_1, \beta_2, \ldots, \beta_{r-1})$, kernel parameters ($\sigma_f^2$ and κ), and noise parameter (σ_n^2). The parameters σ_f^2, κ, and σ_n^2 appear in (9) through the expressions for mean (μ_*) and variance (σ_*^2). Estimating the optimal model parameters (θ^*) (model selection) can be done using the leave-one-out cross-validation (LOO-CV) technique which we will discuss in Section 4.1. The prediction is made by selecting the ordinal category with highest probability, *i.e.* $\underset{1 \le k \le r}{\operatorname{argmax}}\, p(y_* = k | x_*, \mathbf{X}, \mathbf{y}, \theta^*)$.

4.1 Model Selection Using Leave-One-Out Cross-Validation

Model selection for the PLSOR approach is done using the leave-one-out cross-validation (LOO-CV) [7] technique. The log predictive probability of the i^{th} training example x_i, when learnt using the remaining training examples, is

$$\log p(y_i | \mathbf{X}, \mathbf{y_{-i}}, \theta) = \log\left(\Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i}}{\sqrt{1 + \alpha^2\sigma_{-i}^2}} \right) - \Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i - 1}}{\sqrt{1 + \alpha^2\sigma_{-i}^2}} \right) \right) \quad (10)$$

where $y_i \in \{1, 2, \ldots, r\}$ is the output of i^{th} training example x_i and $\mathbf{y_{-i}}$ is the output vector of the remaining training examples. The predictive distribution mean μ_{-i} and variance σ_{-i}^2 for the training example x_i are obtained by performing a Gaussian process regression on all training examples except x_i and are given by (4) and (5) respectively. Model parameters (θ) are estimated by optimizing the sum of the log leave-one-out (LOO) predictive probability (10) over all the training examples. The optimization problem is defined as follows

$$\begin{aligned}
(\theta^*) &= \arg\min_{\theta} \mathcal{L}(\theta) = \arg\min_{\theta} - \sum_{i=1}^{n} \log p(y_i | \mathbf{X}, \mathbf{y_{-i}}, \theta) = \\
&\underset{\alpha, \beta_1, \Delta_2, \ldots, \Delta_{r-1}, \kappa, \sigma_f^2, \sigma_n^2}{\arg\min} - \sum_{i=1}^{n} \log\left(\Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i}}{\sqrt{1 + \alpha^2\sigma_{-i}^2}} \right) - \Phi\left(\frac{\alpha\mu_{-i} + \beta_{y_i - 1}}{\sqrt{1 + \alpha^2\sigma_{-i}^2}} \right) \right) \\
&\text{subject to} \quad \beta_1 \in \mathcal{R}, \\
&\beta_j = \beta_1 + \sum_{l=2}^{j} \Delta_l \quad \forall j = 2, \ldots, r-1 \quad , \quad \Delta_l \ge 0 \quad \forall l = 2, \ldots, r-1. \qquad (11)
\end{aligned}$$

This problem minimizes the negative log predictive probability (NLP) measure over all the training examples. Note that the constraint, $\beta_1 \le \beta_2 \le \ldots \le \beta_{r-1}$, is imposed by redefining the threshold variables as $\beta_j = \beta_1 + \sum_{l=2}^{j} \Delta_l$ using positive padding variables Δ_l. The optimal model parameter values are obtained by solving the optimization problem (11). The optimal model parameter values are used to make prediction using (9).

The proposed approach requires the computation of predictive mean and variance for n training examples. Computation of the predictive mean and variance for each training example involves inversion of an $(n-1) \times (n-1)$ covariance

matrix which requires $\mathcal{O}(n^3)$ time. Therefore the complete LOO-CV procedure takes $\mathcal{O}(n^4)$ time which makes the method computationally expensive. But we get around this problem by noting that we need to perform inversion of only one covariance matrix, covariance matrix $\mathbf{K}$, formed by all training examples. It is then used to compute the predictive mean μ_{-i} and variance σ^2_{-i} for each leave-one-out case as [10]

$$\mu_{-i} = y_i - \left[\mathbf{K}^{-1}\mathbf{y}\right]_i / \left[\mathbf{K}^{-1}\right]_{ii} \tag{12}$$

$$\sigma^2_{-i} = 1/\left[\mathbf{K}^{-1}\right]_{ii}. \tag{13}$$

To evaluate the expressions for μ_{-i} and σ^2_{-i}, we need to perform inversion of the covariance matrix $\mathbf{K}$ and it takes $\mathcal{O}(n^3)$ time. Once we have $\mathbf{K}^{-1}$, we precompute $\mathbf{K}^{-1}\mathbf{y}$ and the computation of μ_{-i} and σ^2_{-i} for the leave-one-out case i is done in constant time using (12) and (13) respectively. The computational complexity of the entire LOO-CV procedure is dominated by the covariance matrix inversion and it is $\mathcal{O}(n^3)$.

The proposed approach, PLSOR, provides a simple and straightforward way to perform ordinal regression using Gaussian processes. In PLSOR, the model parameters are learnt using LOO-CV technique which is easier to implement than the Bayesian techniques employed in MAP-GPOR or EP-GPOR. The entire LOO-CV procedure takes $\mathcal{O}(n^3)$ time, and hence the computational complexity of PLSOR is the same as that of MAP-GPOR or EP-GPOR. In PLSOR, the predictive distribution of test latent functions is learnt using GPR, which in turn uses the likelihood (2) for the training outputs, while the predictive distribution of the test outputs is learnt using the likelihood (7). We call the former likelihood as the training likelihood and the latter as the test likelihood. PLSOR differs from MAP-GPOR or EP-GPOR in using distinct training and test likelihoods. Further PLSOR does not use any approximations unlike MAP-GPOR or EP-GPOR. A summary of the Gaussian process approaches to ordinal regression, MAP-GPOR, EP-GPOR and PLSOR, is given in Table 1.

Table 1. A Summary of the properties of the Gaussian process approaches to ordinal regression, MAP-GPOR, EP-GPOR and PLSOR

Property	MAP-GPOR	EP-GPOR	PLSOR
Training likelihood	$\Phi\left(\frac{b_y - f}{\sigma}\right) - \Phi\left(\frac{b_{y-1} - f}{\sigma}\right)$	$\Phi\left(\frac{b_y - f}{\sigma}\right) - \Phi\left(\frac{b_{y-1} - f}{\sigma}\right)$	$\mathcal{N}(f, \sigma_n^2)$
Test likelihood	$\Phi\left(\frac{b_y - f}{\sigma}\right) - \Phi\left(\frac{b_{y-1} - f}{\sigma}\right)$	$\Phi\left(\frac{b_y - f}{\sigma}\right) - \Phi\left(\frac{b_{y-1} - f}{\sigma}\right)$	$\Phi(b_y - (\alpha f + \beta)) - \Phi(b_{y-1} - (\alpha f + \beta))$
Inference	Laplace approximation	Expectation propagation approximation	Exact, no approximation
Model selection	Evidence maximization	Evidence maximization	NLP minimization
Computational complexity	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$	$\mathcal{O}(n^3)$

5 Experimental Results

We perform the experiments on synthetic and benchmark data sets to compare the performance of the proposed PLSOR approach with MAP-GPOR and EP-GPOR approaches. First, we conduct experiments on the synthetic data set to visualize the behavior of the approaches and then, we study their generalization performance on several benchmark data sets.

5.1 Synthetic Data

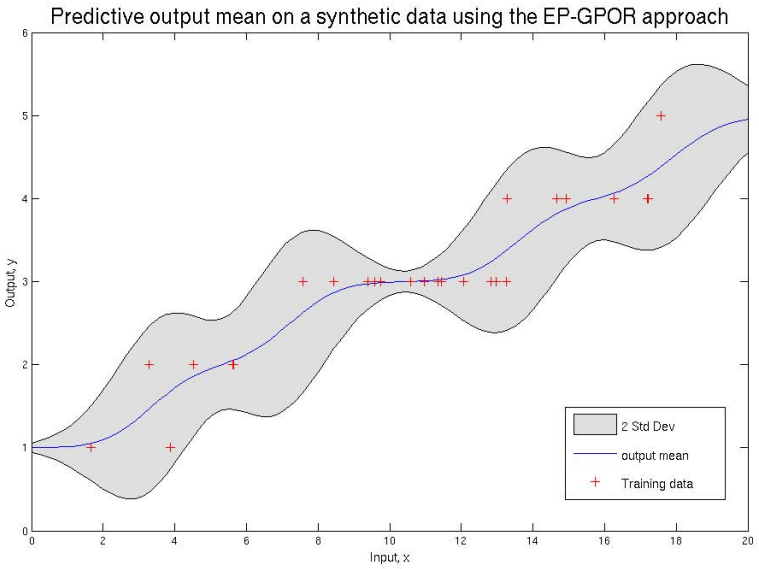
We conduct experiments on a 1-dimensional synthetic data set with five ordinal categories. The training data set contains 20 points (marked by pluses in Fig.1) with two training data points in the interval $[2, 4]$ belonging to category 1, three in the interval $[4, 6]$ belonging to category 2, ten in the interval $[8, 14]$ belonging to category 3, five in the interval $[14, 18]$ belonging to category 4, and one in the interval $[18, 20]$ belonging to category 5. The test data consists of 200 points in the interval $[0, 20]$, each separated by a distance of 0.1. Fig.1(a) shows the mean and the confidence bound of the output predictive distribution for EP-GPOR on the synthetic data set. Similar plot for PLSOR is depicted in Fig.1(b). From Fig.1, we observe that the performance of both the approaches is similar.

5.2 Benchmark Data

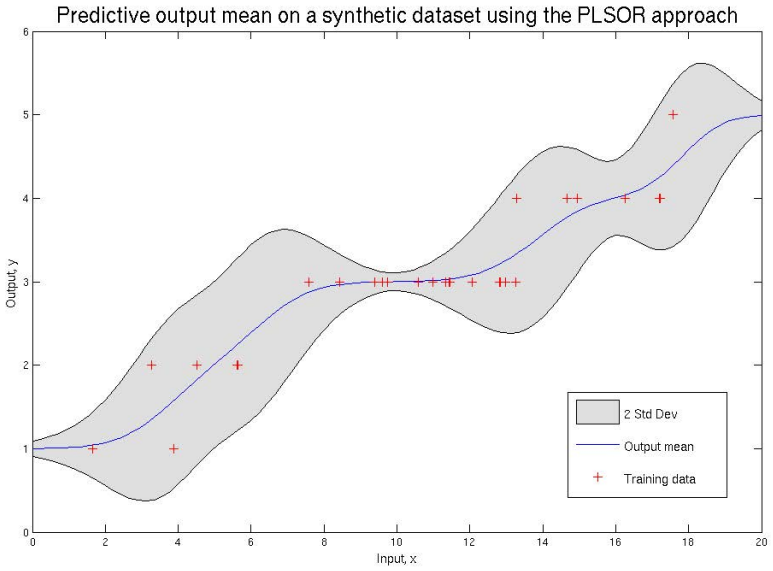
We report the experimental results of our approach on 9 benchmark data sets [8]. Properties of these benchmark data sets are summarized in Table 2. These are regression data sets. The continuous target values are discretized into ordinal values using equal frequency binning. Here, we divide the range of target values into intervals of the same length. Target values are then relabeled according to the interval in which they fall. The ordinal target values thus obtained range from 1 to r, where r denotes the number of intervals. For each data set, we generate two versions, 5 bins and 10 bins, obtained by discretizing the target values in the original data set into 5 and 10 intervals respectively. We conduct experiments on both the versions of the data sets. Each data set is randomly

Table 2. Benchmark data sets and their properties

Data set	Attributes	Training Instances	Test Instances
Diabetes	2	30	13
Pyrimidine	27	50	24
Triazines	60	100	86
Wisconsin	32	130	64
Machine	6	150	59
AutoMPG	7	200	192
Boston	13	300	206
Stocks	9	600	350
Abalone	8	1000	3177



(a) EP-GPOR



(b) PLSOR

Fig. 1. The mean value and the confidence bound of the output predictive distribution for EP-GPOR and PLSOR on an 1-dimensional synthetic data set

Table 3. Comparison of the results of PLSOR with MAP-GPOR and EP-GPOR on benchmark data sets for the 5 bins version. Mean zero-one errors are reported in percentage. Mean absolute errors are rounded off to 2 decimal places. Values in bold letters denote the lowest mean value among the three approaches.

	Mean zero-one error(%)			Mean absolute error		
Data	MAP-GPOR	EP-GPOR	PLSOR	MAP-GPOR	EP- GPOR	PLSOR
Diabetes	54.23±13.78	54.23±13.78	**48.46±11.2**	0.66±0.14	0.67±0.14	**0.62±0.16**
Pyrimidine	39.79±7.21	**36.46±6.47**	39.37±9.41	0.43±0.09	**0.39±0.07**	0.46±0.19
Triazines	52.91±2.15	**52.62±2.66**	54.42±3.43	**0.69±0.02**	0.69±0.03	0.74±0.063
Wisconsin	**65.00±4.71**	65.16±4.65	65.70±3.23	**1.01±0.09**	1.01±0.09	1.24±0.10
Machine	**16.53±3.56**	16.78±3.88	18.39±3.45	**0.19±0.04**	0.19±0.04	0.21±0.05
AutoMPG	23.78±1.85	**23.75±1.74**	25.76±2.19	0.24±0.02	**0.24±0.02**	0.26±0.02
Boston	24.88±2.02	**24.49±1.85**	24.59±2.57	0.26±0.02	0.26±0.02	**0.26±0.02**
Stocks	11.99±2.34	12.00±2.06	**10.70±1.66**	0.12±0.02	0.12±0.02	**0.11±0.02**
Abalone	**21.50±0.22**	21.56±0.36	22.05±0.30	**0.23±0.00**	0.23±0.00	0.24±0.00

partitioned into training and test data sets and 20 such training and test data set instances are generated by repeated independent partitioning. We use the Gaussian kernel (1) in all our experiments.

The model parameter values are obtained by solving the optimization problem (11). The optimization is run with random as well as fixed[1] initialization of optimization variables; we report the result for which the objective function value is the least.

We compare the generalization performance of PLSOR with MAP-GPOR and EP-GPOR on the benchmark datasets. We use two evaluation metrics to compare the performance, *zero-one error* and *absolute error* [8]. Let the actual test outputs be $\{y_1, \ldots, y_{n_*}\}$ and the predicted test outputs be $\{\hat{y_1}, \ldots, \hat{y_{n_*}}\}$. Then the *zero-one error* and *absolute error* are defined as follows.

zero-one error. gives the fraction of incorrect predictions on test data *i.e.* $\frac{1}{n_*} \sum_{i=1}^{n_*} \mathbb{I}(\hat{y_i} \neq y_i)$, where $\mathbb{I}(\cdot)$ is an indicator function which gives 1 when the argument is true and 0 otherwise.

absolute error. gives the average deviation of predicted test outputs from the actual test outputs *i.e.* $\frac{1}{n_*} \sum_{i=1}^{n_*} |\hat{y_i} - y_i|$.

For each data set, zero-one and absolute errors for the proposed approach is obtained on all the 20 instances of training and test data sets. The mean of the zero-one and absolute errors, along with their standard deviation, are used to compare the performance of various approaches. We prefer methods with low mean zero-one and mean absolute errors. Tables 3 and 4 compare PLSOR with MAP-GPOR and EP-GPOR for the 5 bins and 10 bins cases respectively.

We observe from Tables 3 and 4 that the results obtained with the PLSOR approach are comparable with those obtained with the MAP-GPOR and

[1] Fixed initialization is done as given in [8] where we choose $\sigma_f^2 = 1, \kappa = 1/d$, d being the dimension of the data set, $\beta_1 = -1, \Delta_l = 2/r$, r being number of ordinal categories.

Table 4. Comparison of the results of PLSOR with MAP-GPOR and EP-GPOR on benchmark data sets for the 10 bins version. Mean zero-one errors are reported in percentage. Mean absolute errors are rounded off to 2 decimal places. Values in bold letters denote the lowest mean value among the three approaches.

	Mean zero-one error(%)			Mean absolute error		
Data	MAP-GPOR	EP-GPOR	PLSOR	MAP-GPOR	EP-GPOR	PLSOR
Diabetes	83.46±5.73	83.08±5.91	**76.92±9.98**	2.14±0.33	2.14±0.33	**1.50±0.37**
Pyrimidine	55.42±8.01	**54.38±7.70**	55.63±8.47	0.88±0.18	**0.83±0.13**	0.89±0.18
Triazines	**63.72±4.34**	64.01±3.78	69.88±4.97	**1.20±0.07**	1.20±0.07	1.37±0.20
Wisconsin	78.52±3.58	78.52±3.51	**75.94±1.86**	**2.14±0.18**	2.14±0.18	2.94±0.13
Machine	33.81±3.91	**33.73±3.64**	35.17±3.64	0.48±0.07	**0.47±0.08**	0.53±0.08
Auto MPG	43.96±2.81	**43.88±2.60**	46.35±2.48	0.50±0.03	**0.50±0.03**	0.56±0.04
Boston	41.53±2.77	**41.26±2.86**	41.99±2.82	0.49±0.03	**0.49±0.03**	0.51±0.04
Stocks	19.90±1.72	19.44±1.91	**18.17±1.79**	0.20±0.02	0.20±0.02	**0.19±0.02**
Abalone	42.60±0.91	**42.27±0.46**	44.24±0.68	0.51±0.01	**0.51±0.01**	0.55±0.01

Table 5. Average rank of each of the ordinal regression approaches, MAP-GPOR, EP-GPOR and PLSOR, over all the data sets and the Friedman statistic computed over all the approaches

	5 bins		10 bins	
	zero-one	absolute	zero-one	absolute
MAP-GPOR	1.944	1.778	2.167	1.833
EP-GPOR	1.722	1.778	1.500	1.611
PLSOR	2.333	2.444	2.333	2.556
F_F	0.8266	1.3880	1.8449	2.5841

EP-GPOR approaches. PLSOR is found to perform better than MAP-GPOR and EP-GPOR on two data sets, Diabetes and Stocks. On other data sets, the PLSOR results are close to the MAP-GPOR and EP-GPOR results.

We use the Friedman test [11] to check if the performance of the proposed approach differs significantly from the existing GPOR approaches. Here we compare 3 approaches on 9 data sets. Therefore, the F distribution has 2 and 16 degrees of freedom[2]. For the level of significance $\alpha = 0.05$, the critical F value is 3.63. Table 5 reports the average rank of the ordinal regression approaches, MAP-GPOR, EP-GPOR and PLSOR, over all the data sets. It also reports the Friedman statistic F_F [11] computed over all approaches for 5 bins and 10 bins cases with respect to zero-one and absolute errors. In all the cases, the computed F_F values are less than the critical F value (due to the ranks being similar). Hence there does not exist any significance differences between various approaches. Thus, the proposed PLSOR approach is simple, easy to implement and gives competitive performance compared to the existing state-of-the-art GP based approaches for ordinal regression.

[2] For K approaches and N data sets, F distribution has $K-1$ and $(K-1)(N-1)$ degrees of freedom.

6 Conclusion

In this work, we proposed a novel approach to solve the ordinal regression problem using Gaussian processes. The proposed approach, probabilistic least squares ordinal regression (PLSOR), provided an easy and exact way to perform ordinal regression using Gaussian processes. Here model selection is performed using leave-one-out cross-validation technique. Experiments on synthetic and benchmark data sets showed that the proposed approach is competitive with the state-of-the-art GPOR approach. In future, we would like to develop sparse models for the Gaussian process ordinal regression approaches so that the training time and inference time could be reduced considerably.

References

1. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Advances in Large Margin Classifiers. MIT Press (2000)
2. Shashua, A., Levin, A.: Ranking with Large Margin Principle: Two Approaches. In: Advances in Neural Information Processing Systems 15, pp. 937–944. The MIT Press (2003)
3. Chu, W., Keerthi, S.S.: New Approaches to Support Vector Ordinal Regression. In: Proceedings of the 22nd International Conference on Machine Learning, pp. 145–152. ACM (2005)
4. Sun, B.Y., Li, J., Wu, D.D., Zhang, X.M., Li, W.B.: Kernel Discriminant Learning for Ordinal Regression. IEEE Trans. on Knowl. and Data Eng. 22, 906–910 (2010)
5. Chang, X., Zheng, Q., Lin, P.: Ordinal Regression with Sparse Bayesian. In: Huang, D.-S., Jo, K.-H., Lee, H.-H., Kang, H.-J., Bevilacqua, V. (eds.) ICIC 2009. Part II. LNCS, vol. 5755, pp. 591–599. Springer, Heidelberg (2009)
6. McCullagh, P.: Regression Models for Ordinal Data. Journal of the Royal Statistical Society 42, 109–142 (1980)
7. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
8. Chu, W., Ghahramani, Z.: Gaussian Processes for Ordinal Regression. J. Mach. Learn. Res. 6, 1019–1041 (2005)
9. Minka, T.: A Family of Algorithms for Approximate Bayesian Inference. PhD thesis, Massachusetts Institute of Technology (2001)
10. Sundararajan, S., Keerthi, S.S.: Predictive Approaches for Choosing Hyperparameters in Gaussian Processes. Neural Computation 13, 1103–1118 (1999)
11. Demsar, J.: Statistical Comparisons of Classifiers over Multiple Data Sets. J. Mach. Learn. Res. 7, 1–30 (2006)