Semi-supervised Gaussian Process 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] Microsoft Research, Bangalore, India
ssrajan@microsoft.com

Abstract. Ordinal regression problem arises in situations where examples are rated in an ordinal scale. In practice, labeled ordinal data are difficult to obtain while unlabeled ordinal data are available in abundance. Designing a probabilistic semi-supervised classifier to perform ordinal regression is challenging. In this work, we propose a novel approach for semi-supervised ordinal regression using Gaussian Processes (GP). It uses the expectation-propagation approximation idea, widely used for GP ordinal regression problem. The proposed approach makes use of unlabeled data in addition to the labeled data to learn a model by matching ordinal label distributions approximately between labeled and unlabeled data. The resulting mixed integer programming problem, involving model parameters (real-valued) and ordinal labels (integers) as variables, is solved efficiently using a sequence of alternating optimization steps. Experimental results on synthetic, bench-mark and real-world data sets demonstrate that the proposed GP based approach makes effective use of the unlabeled data to give better generalization performance (on the absolute error metric, in particular) than the supervised approach. Thus, it is a useful approach for probabilistic semi-supervised ordinal regression problem.

Keywords: Gaussian processes, ordinal regression, semi-supervised learning, annealing

1 Introduction

We consider the problem of predicting variables of ordinal scale, a setting referred to as *ordinal regression*. These problems arise in many different domains like Social Sciences, Bioinformatics and Information Retrieval. For example, a user can label a retrieved document using one of the following categories: *highly relevant, relevant, average, irrelevant* and *highly irrelevant*. There exists an order among the labels, which makes the ordinal regression problems different from classification problems. Further, the labels are discrete and not continuous, unlike in the regression problems.

Although the problem of ordinal regression is well studied in Statistics [1–3], there has been a surge of interest, in recent years, in solving this problem in a

learning framework. The ordinal regression problem can be solved by treating it as a regression problem after transforming the ordinal scales into numeric values [4], or by converting it into nested binary classification problems that encode the ordering of the original ranks [5]. This solution strategy can be referred to as a reduction framework. Alternatively, the problem can be solved directly using machine learning algorithms like support vector machines (SVM) [6] or Gaussian Processes (GP) [7].

In many practical applications, labeled data are scarce to obtain. For example, in the domain of Bioinformatics, time consuming experiments and domain knowledge (biological experts) are required to label the data. Thus, obtaining the label information is expensive and time consuming. However, unlabeled data are easily available and are present in abundance. *Semi-supervised learning* [8] uses the unlabeled data along with the labeled data to learn better predictive models. Many approaches have been developed for the semi-supervised learning of regression and classification tasks. These approaches are based on various assumptions on the unlabeled data like clustering, smoothness or manifold [8]. They can be broadly classified as generative approaches, graph based approaches and approaches implementing low-density separation [8]. There exists a rich literature on semi-supervised regression and classification. See [8] and the references therein for more details. However, there is not much work reported in the literature to solve semi-supervised ordinal regression problem.

Semi-supervised ordinal regression problems arise quite naturally in several contexts. For instance, in recommendation systems, every user may rate only a few items. Often, the labeled ordinal data are insufficient to learn a good ordinal regression model. Most of the literature on ordinal regression [6, 7, 9–12] focused on the supervised learning setting. Recently, transductive ordinal regression (TOR) [13] approach was proposed to perform ordinal regression in a semi-supervised setting. The approach uses the reduction framework to solve the ordinal regression problem and learns the labels of the unlabeled examples and the decision function iteratively. The approach can be used for a general class of loss functions and was shown to give better performance than the approach which used only labeled examples. Semi-supervised manifold ordinal regression [14] is a new approach for semi-supervised ordinal regression for image ranking. This approach uses the assumption that is most appropriate for image analysis: the high dimensional observations lie on or close to a low-dimensional manifold. However, none of these approaches offer a solution to the semi-supervised ordinal regression problem in the Bayesian setting.

In the Bayesian setting, Bayesian committee machine [15] is one of the early attempts to solve a transductive regression problem using Gaussian processes. Though computationally expensive, it performs well on low noise data sets. Null category noise model [16] provides a semi-supervised approach to Gaussian process classification. A disadvantage of this approach is that the Gaussian approximation to the noise model can have negative variance. Semi-supervised Gaussian process classifiers [17] use a graph based approach to learn semi-supervised GP classifiers. It is based on using geometric properties of unlabeled

data within globally defined kernel functions. It is extended to regression problems in [18]. They also propose a feedback mechanism in which the model is retrained by considering some unlabeled data and its predictions as labeled data. The Archipelago model [19] presents a generative approach for semi-supervised GP classification. It uses a GP to specify priors over label distribution and uses it along with a base distribution to model data distribution. More closely related to our work is the "Distribution Matching" approach for transductive regression and classification [20]. This approach is designed for a large margin setting. In a GP setting, similar ideas are used in [21] and [22] for transductive GP regression and multi-category classification, respectively. However, none of these transductive or semi-supervised GP based approaches are extended to semi-supervised ordinal regression problem.

Contributions: We propose a novel approach for semi-supervised ordinal regression using Gaussian Processes. GPs are non-parametric Bayesian models and provide a probabilistic kernel based approach for learning. Our method, hereafter abbreviated as SSGPOR, learns decision boundaries which pass through a low density region. The proposed approach is based on the assumption that the output distributions corresponding to labeled and unlabeled data are similar, a well founded assumption explored in the transductive classification and regression settings [20]. The proposed approach models the similarity by minimizing the Kullback-Leibler (KL) divergence between the predictive distribution over the unlabeled data outputs and an approximate distribution . The approximate distribution has properties similar to the labeled data output distribution. Obtaining the approximate distribution satisfying these properties is challenging. Our approach involves solving two sub-problems iteratively: (1) We learn the model by minimizing an upper bound on the negative logarithm of the evidence and the KL divergence, (2) we estimate the approximate distribution efficiently using the label switching method [23] that solves an underlying integer programming problem. To avoid bad local minima that typically arise with the unlabeled data in the semi-supervised setting, we use an annealing technique where the contribution of the unlabeled loss term is gradually increased [24].

Our method can be seen as an extension of the supervised Gaussian process ordinal regression approach using expectation propagation (EPGPOR) [7], to the semi-supervised setting. The EPGPOR approach is among the state-of-the-art approaches for ordinal regression. We compare the performance of the proposed SSGPOR approach with the EPGPOR approach. The experiments on synthetic, benchmark and real-world data sets show that, the performance of the EPGPOR approach could be significantly improved using our method when unlabeled data are available. It is also observed that the SSGPOR approach performs better than the TOR approach [13] in the transductive setting. Large improvements are observed on the absolute error metric than zero-one error metric. Note that unlike classification problems where zero-one error is important, absolute error metric is more meaningful in ordinal regression problems.

The rest of the paper is organized as follows. In Sect. 2, we introduce the Gaussian process and discuss the Gaussian process ordinal regression approach

using expectation propagation (EPGPOR). Section 3 discusses the proposed approach, semi-supervised Gaussian process ordinal regression (SSGPOR), in detail. Comparisons of the SSGPOR, EPGPOR and TOR approaches on synthetic, benchmark and real-world data sets are presented in Sect. 4. Finally, some conclusions are drawn in Sect. 5.

We use the following notations for the discussion ahead. Given a sample of n_l labeled independent examples $\mathcal{D}_l = (X_l, \mathbf{y}_l) = \{(\mathbf{x}_i, y_i)\}_{i=1}^{n_l}$ and n_u unlabeled independent examples $\mathcal{D}_u = (X_u) = \{(\mathbf{x}_i)\}_{i=1}^{n_u}$. Let $\mathcal{D} = \mathcal{D}_l \cup \mathcal{D}_u$ denote the set of all training examples of size n ($n = n_l + n_u$). Let $\mathcal{D}_*$ be the set consisting of n_* test data points X_*. We assume $\mathbf{x}_i \in \mathbb{X} \subseteq \mathbb{R}^d$ and $y_i \in \mathbb{Y} = \{c_1, c_2, \ldots, c_r\}$, where $c_1 < c_2 < \ldots < c_r$. We consider an ordinal regression problem with r ordered categories and without loss of generality, we denote them by r consecutive integers $\{1, 2, \ldots, r\}$. Our goal is to learn a decision function $h : \mathbb{X} \to \mathbb{Y}$ from both labeled and unlabeled data, such that it generalizes well on test data.

2 Background

A Gaussian process (GP) is a collection of random variables with the property that the joint distribution of any finite subset of the variables is a Gaussian [25]. It generalizes the Gaussian distribution to infinitely many random variables. The GP is used to define a prior distribution over latent functions underlying a model. It is completely specified by a mean function and a covariance function. The covariance function is defined over latent function values of a pair of input examples and is typically evaluated using the Mercer kernel function over the pair of input examples. The covariance function expresses some general properties of functions such as their smoothness, and length-scale. A commonly used covariance function is the squared exponential (SE) or the Gaussian kernel

$$cov\big(t_i, t_j\big) = k(\mathbf{x}_i, \mathbf{x}_j) = \exp(-\frac{\kappa}{2}\|\mathbf{x}_i - \mathbf{x}_j\|^2). \tag{1}$$

Here $t_i = t(\mathbf{x}_i)$ and $t_j = t(\mathbf{x}_j)$ are latent function values associated with the inputs $\mathbf{x}_i$ and $\mathbf{x}_j$ respectively. $\kappa > 0$ is the hyper-parameter associated with the covariance function and $\|\cdot\|$ is the L_2 norm. The latent function sampled from a GP is denoted by t and in particular we denote the latent functions associated with labeled data as $\mathbf{t_l}$, unlabeled data as $\mathbf{t_u}$ and test data as $\mathbf{t}_*$. Let $K_{ll} = k(X_l, X_l)$, $K_{l*} = k(X_l, X_*)$ and $K_{**} = k(X_*, X_*)$. Here $k(X_l, X_*)$ is an $n_l \times n_*$ matrix of covariances evaluated at all pairs of labeled training and test input data. The matrices $k(X_l, X_l)$, $K(X_*, X_l)$ and $K(X_*, X_*)$ are defined similarly.

Gaussian Process Ordinal Regression The Gaussian process ordinal regression (GPOR) [7] approach uses a non Gaussian likelihood function for modeling the ordinal labels. It uses a zero mean Gaussian process prior on the latent function values $t(\mathbf{x})$. Under noisy observations, for an input $\mathbf{x}$, the likelihood function for an ordinal output y is defined as

$$p(y|t(\mathbf{x})) = \Phi\big(\frac{b_y - t(\mathbf{x})}{\sigma}\big) - \Phi\big(\frac{b_{y-1} - t(\mathbf{x})}{\sigma}\big), \tag{2}$$

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 (2) is not a Gaussian and therefore the posterior, $p(\mathbf{t_l}|\mathcal{D}_l)$, could not be obtained in closed form. The GPOR approach works by approximating the posterior as a Gaussian distribution using either Laplace approximation (MAPGPOR) or using expectation propagation (EPGPOR).

Learning The Expectation propagation (EP) [26] approach approximates the posterior $p(\mathbf{t_l}|\mathcal{D}_l) \propto \prod_{i=1}^{n_l} p(y_i|t_i)p(\mathbf{t_l})$ as a product of Gaussian distributions $r(\mathbf{t_l}; \mathbf{h}, A) = \prod_{i=1}^{n_l} \hat{p}(t_i)p(\mathbf{t_l})$, where $\hat{p}(t_i) = s_i \exp(-\frac{1}{2}p_i(t_i - m_i)^2)$, $A = (K_{ll}^{-1} + \Pi)^{-1}$, and $\mathbf{h} = A\Pi\mathbf{m}$. Here, Π is a $n_l \times n_l$ diagonal matrix with elements in the diagonal given by $\{p_i\}_{i=1}^{n_l}$ and $\mathbf{m}$ is a n_l dimensional column vector with elements given by $\{m_i\}_{i=1}^{n_l}$. The parameters $\{s_i, m_i, p_i\}_{i=1}^{n_l}$ are called the site parameters of the EP approximation. The site parameters are obtained iteratively where in each iteration i, $\{s_i, m_i, p_i\}$ are obtained by minimizing the Kullback-Leibler (KL) divergence [8], $KL(r_{-i}(t_i)p(y_i|t_i) \,||\, r_{-i}(t_i)\hat{p}(t_i))$. Here $r_{-i}(t_i)$ is the marginal cavity distribution over t_i obtained after leaving out the i^{th} likelihood term $\hat{p}(y_i|t_i)$ from the approximated posterior $r(\mathbf{t_l})$ and then marginalizing over the remaining variables.

The EPGPOR approach performs model selection by minimizing an upper bound ($\mathcal{F}(\theta)$) on the negative logarithm of evidence ($p(\mathcal{D}_l|\theta)$) ,

$$\begin{aligned}\underset{\theta}{\operatorname{argmin}}\, \mathcal{F}(\theta) = \underset{\theta}{\operatorname{argmin}} - \sum_{i=1}^{n_l} \int r(t_i; h_i, A_{ii}) \log(\phi(\frac{b_{y_i} - t_i}{\sigma}) - \phi(\frac{b_{y_i-1} - t_i}{\sigma}))dt_i \\ + \frac{1}{2} log|I + K_{ll}\Pi| + \frac{1}{2} tr(I + K_{ll}\Pi)^{-1} + \frac{1}{2}\mathbf{m}^\top (K_{ll} + \Pi^{-1})^{-1} K_{ll} (K_{ll} + \Pi^{-1})^{-1}\mathbf{m}\end{aligned} \tag{3}$$

where θ is the model parameter vector which includes the kernel parameter κ in the covariance function, the threshold parameters $(b_1, b_2, \ldots, b_{r-1})$ and the noise parameter σ in the likelihood function. Here, $tr(B)$ denotes the trace of the matrix B. The optimization can be done using any standard gradient based techniques like conjugate gradient. During optimization, for every new model parameter values, the site parameters and the approximated posterior $r(\mathbf{t_l})$ are re-estimated using the EP approach.

Prediction The learnt model parameters and the EP approximated posterior are used to make predictions on test data. The predictive distribution of the latent function t_* for a test data $\mathbf{x}_*$ is $p(t_*|\mathbf{x}_*, \mathcal{D}_l) \sim N(t_*; \mu_*, \sigma_*^2)$, where $\mu_* = K_{l*}^\top (K_{ll} + \Pi^{-1})^{-1}\mathbf{m}$ and $\sigma_*^2 = K_{**} - K_{l*}^\top (K_{ll} + \Pi^{-1})^{-1} K_{l*}$. The predictive distribution for test output is $p(y_*|\mathbf{x}_*, \mathcal{D}_l) = \phi\big(\frac{b_{y_*} - \mu_*}{\sqrt{\sigma^2 + \sigma_*^2}}\big) - \phi\big(\frac{b_{y_*-1} - \mu_*}{\sqrt{\sigma^2 + \sigma_*^2}}\big)$.

The EPGPOR approach is a supervised approach. It does not perform well when the size of the labeled data are small. In most of the practical scenarios, labeled data are limited while unlabeled data are available in abundance. We propose a semi-supervised approach which extends the EPGPOR approach to a semi-supervised setting. The proposed approach make use of the unlabeled data along with the labeled data to learn a better decision function than the EPGPOR approach.

3 Semi-supervised Gaussian Process Ordinal Regression

The proposed approach, semi-supervised Gaussian process ordinal regression (SSGPOR), is based on the idea of "Distribution Matching" [20–22] and is derived by extending the transductive GP regression (TGPR) [21] approach to the ordinal regression setting. The basic assumption is that the predictive distribution on unlabeled data should have properties similar to the output distribution on labeled data. In particular, it requires the average number of examples for an ordinal category in unlabeled data should match approximately with the average number of examples for that category in labeled data. The assumption is justified by the independent and identically distributed (i.i.d.) nature of the data and is true for many real-world data sets [21]. The model parameters are estimated subject to these assumptions. It results in distributions which are consistent across labeled and unlabeled data. We now briefly describe the TGPR approach and then explain the proposed approach in detail.

The TGPR approach [21] models the regression problem where the output is real valued and the likelihood is a Gaussian. It considers a transductive setting where the training data set is $\mathcal{D}_l \cup \mathcal{D}_u$ and the designed GP model is used to predict the labels of the examples in $\mathcal{D}_u$. The TGPR approach requires the predictive Gaussian distribution over unlabeled data to be close to a family of Gaussian distributions $\hat{Q}$. The family $\hat{Q}$ is such that the first and second moments of its members on unlabeled data are close to the corresponding moments obtained using labeled data. The model parameters ($\hat{\theta}$) are obtained by minimizing the negative logarithm of evidence ($p(\mathcal{D}_l|\hat{\theta})$), subject to the constraint that the predictive distribution over unlabeled data $p(\mathbf{y}_u|\mathcal{D}_l, \mathcal{D}_u, \hat{\theta})$, belongs to the approximating family $\hat{Q}$. The constraint could be enforced by minimizing the Kullback-Leibler (KL) divergence between $p(\mathbf{y}_u|\mathcal{D}_l, \mathcal{D}_u, \hat{\theta})$ and some $\hat{q} \in \hat{Q}$ [21]. The model parameters ($\hat{\theta}$) and $\hat{q} \in \hat{Q}$ are estimated by solving the joint optimization problem ;

$$\underset{\hat{q}\in\hat{Q},\hat{\theta}}{\text{argmin}} \quad -\log p(\mathcal{D}_l|\hat{\theta}) + \lambda\, KL(\hat{q}(\mathbf{y}_u)||p(\mathbf{y}_u|\mathcal{D}_l, \mathcal{D}_u, \hat{\theta})). \tag{4}$$

Here, λ is a regularization parameter and for two distributions q and p, $KL(q||p) = \int q(y) \log \frac{q(y)}{p(y)} dy$. The parameters are obtained using an alternating optimization approach [21].

It is not easy to extend the TGPR approach to the ordinal regression setting. This is due to the nature of the labels and the likelihood. In ordinal regression,

the labels are discrete and ordered. Further, the likelihood is non-Gaussian. Since labels are discrete and ordered, we have to consider a discrete approximating distribution. Because of the non-Gaussian nature of the likelihood, we have to use approximation techniques like expectation propagation to obtain a Gaussian approximated posterior [7]. The discrete nature of the labels results in an integer programming problem which needs to be solved efficiently. We now give the details of the proposed approach.

Proposed Approach The SSGPOR approach considers the setting where the training data set is $\mathcal{D}_l \cup \mathcal{D}_u$ and the designed GP model is tested on $\mathcal{D}_*$. It uses the likelihood (2) and the expectation propagation approach [7], to obtain a Gaussian approximation of the posterior distribution. The resulting predictive distribution on an ordinal output y_u of an unlabeled example $\mathbf{x}_u \in \mathcal{D}_u$ is given as

$$p(y_u|\mathbf{x}_u, \mathcal{D}_l) = \phi\left(\frac{b_{y_u} - \mu_u}{\sqrt{\sigma^2 + \sigma_u^2}}\right) - \phi\left(\frac{b_{y_u-1} - \mu_u}{\sqrt{\sigma^2 + \sigma_u^2}}\right), \;\; y_u = 1, \ldots, r \qquad (5)$$

where $\mu_u = K_{lu}^\top (K_{ll} + \Pi^{-1})^{-1}\mathbf{m}$ and $\sigma_u^2 = K_{uu} - K_{lu}^\top (K_{ll} + \Pi^{-1})^{-1} K_{lu}$.

The SSGPOR approach requires the predictive distribution (5) over the unlabeled data to have some properties similar to the output distribution over the labeled data. We achieve this by considering an approximate distribution over the unlabeled data output with properties similar to the labeled data output distribution, and constrain the predictive distribution to be close to the approximate distribution. Since outputs are discrete in the ordinal regression setting, the approximate distribution takes the form of a multinomial distribution. In particular, we consider a multinomial distribution with r categories such that probability of success, p_j, for each category is defined by the average number of examples of that category in labeled data, *i.e.* $p_j = \gamma_j$, where $\gamma_j = \frac{1}{n_l}\sum_{i=1}^{n_l} \mathbb{I}(y_i = j)$ ($\mathbb{I}(\cdot)$ is an Indicator function). We define a label matrix q of size $n_u \times r$, where each row q_i is an i.i.d. random vector following the multinomial distribution for a single trial and provides a label for the i^{th} unlabeled example. The i^{th} unlabeled example is assigned a label j, if $q_{ij} = 1$. We have $q_{ij} \in \{0, 1\}$ and $\sum_{j=1}^{r} q_{ij} = 1 \;\forall\, i = 1, \ldots, n_u$. Also, q satisfies the label constraints $\frac{1}{n_u}\sum_{i=1}^{n_u} q_{ij} = \gamma_j \;\forall\, j = 1, \ldots, r$, which ensures that the distribution over the unlabeled data are similar to the labeled data distribution. The label constraints are important in a semi-supervised setting as they avoid trivial solutions like assigning all unlabeled data to a single category [8]. Let Q be the set of all q satisfying all these constraints, *i.e.* $Q = \{q : q \in \{0, 1\}^{n_u \times r}, \sum_{j=1}^{r} q_{ij} = 1 \,\forall\, i, \frac{1}{n_u}\sum_{i=1}^{n_u} q_{ij} = \gamma_j \,\forall\, j\}$. The SSGPOR approach requires the predictive distribution over all the unlabeled data $p(\mathbf{y}_u|\mathcal{D}_l, \mathcal{D}_u)$ to be close enough to some $q \in Q$. This can be achieved by minimizing the KL-divergence between q and $p(\mathbf{y}_u|\mathcal{D}_l, \mathcal{D}_u)$. Since obtaining the joint distribution $p(\mathbf{y}_u|\mathcal{D}_l, \mathcal{D}_u)$ is difficult, we instead minimize the sum of the KL divergence between q_u and $p(y_u|\mathbf{x}_u, \mathcal{D}_l)$ over all unlabeled examples.

Objective function The SSGPOR approach estimates the model parameters $\theta = (b_1, b_2, \ldots, b_{r-1}, \kappa, \sigma)$ and $q \in Q$, by minimizing the upper bound on the negative logarithm of evidence (3) and the sum of the KL-divergences over

all unlabeled data. It results in the following joint optimization problem;

$$\underset{q\in Q,\theta}{\operatorname{argmin}} \ \mathcal{F}(\theta) + \lambda \sum_{i=1}^{n_u} KL(q_i||p(y_i|\mathbf{x}_i, \mathcal{D}_l, \theta)). \tag{6}$$

Here, the variable λ serves as a regularization parameter determining the importance that should be given to the unlabeled data term. The model parameters θ and q are obtained by an alternating optimization approach. It is an iterative approach, where in each iteration, we first solve the model parameters keeping q fixed. Then, we estimate $q \in Q$ keeping the model parameters fixed.

Alternating Optimization

(i) Estimating θ For a fixed q, the model parameters (θ) are obtained as

$$\begin{aligned}
&\underset{\theta}{\operatorname{argmin}} \quad \frac{1}{n_l}\mathcal{F}(\theta) - \lambda\frac{1}{n_u}\sum_{i=1}^{n_u} KL(q_i||p(y_i|\mathbf{x}_i, \mathcal{D}_l, \theta)) \\
&= \underset{b_1,\ldots,b_r,\sigma^2,\kappa}{\operatorname{argmin}} -\frac{1}{n_l}\sum_{i=1}^{n_l}\int r(t_i; h_i, A_{ii}) \log(\phi(\frac{b_{y_i} - t_i}{\sigma}) - \phi(\frac{b_{y_i-1} - t_i}{\sigma}))dt_i \\
&-\lambda\frac{1}{n_u}\sum_{i=1}^{n_u}\sum_{j=1}^{r} q_{ij} log(\phi(\frac{b_j - \mu_i}{\sqrt{\sigma^2+\sigma_i^2}}) - \phi(\frac{b_{j-1} - \mu_i}{\sqrt{\sigma^2+\sigma_i^2}})) + \frac{1}{2}log|I + K_{ll}\Pi| \\
&+\frac{1}{2}tr((I + K_{ll}\Pi)^{-1}) + \frac{1}{2}\mathbf{m}^\top(K_{ll} + \Pi^{-1})^{-1}K_{ll}(K_{ll} + \Pi^{-1})^{-1}\mathbf{m} \\
&\text{s.t. } b_1 \leq \ldots \leq b_r
\end{aligned} \tag{7}$$

This problem can be converted to an unconstrained optimization problem and can be solved using any standard optimization technique like conjugate gradient. During optimization, the site parameters and the approximated posterior $r(\mathbf{t_l})$ are re-estimated using the EP approach.

(ii) Estimating q For fixed model parameters, q is estimated by minimizing the sum of the KL-divergences over all the unlabeled data subject to the constraint that $q \in Q$. It results in the following optimization problem.

$$\begin{aligned}
&\underset{q\in\{0,1\}^{n_u\times r}}{\operatorname{argmin}} \quad -\sum_{i=1}^{n_u}\sum_{j=1}^{r} q_{ij} log(\phi(\frac{b_j - \mu_i}{\sqrt{(\sigma^2+\sigma_i^2)}}) - \phi(\frac{b_{j-1} - \mu_i}{\sqrt{(\sigma^2+\sigma_i^2)}})) \\
&\text{s.t. } \frac{1}{n_u}\sum_{i=1}^{n_u} q_{ij} = \gamma_j \ \forall j = 1, \ldots, r \ , \ \sum_{j=1}^{r} q_{ij} = 1 \ \forall i = 1, \ldots, n_u
\end{aligned} \tag{8}$$

Estimation of q is a binary integer programming problem and is done efficiently using the label switching algorithm [23].

We now discuss the proposed SSGPOR algorithm to solve (6) in detail.

Algorithm The SSGPOR algorithm (Algorithm 1) consists of two parts: (i) initialization part (steps 2 and 3) and (ii) iterative part (steps 4–9).

The initialization of model parameters θ (step 2) is done by solving the supervised learning problem using the EPGPOR approach on labeled data, $\mathcal{D}_l$.

It is then used to initialize the label matrix q (step 3) so that constraints are satisfied. This is done as follows. The initialized model parameters are used to find the prediction probability (5) for every category of unlabeled data. For every category, the unlabeled data examples are ranked based on the descending order of their prediction probability for that category. Starting from category 1 to r, the top ranked unlabeled data examples are assigned to the respective categories such that the number of examples assigned to each category does not exceed the expected number $(n_u \times \gamma_j)$. Care should be taken to remove examples from the sorted list corresponding to other categories, once they have been assigned to a particular category.

The iterative part of the algorithm corresponds to solving the problem (6) for different values of the regularization parameter λ. To avoid drastic switching of the labels in q, λ is varied from a small value to a final value 1 in annealing steps. That is, little importance is given to the unlabeled examples in the beginning $(\lambda = 10^{-3})$ and the importance of the unlabeled examples is increased gradually as λ is increased. This helps the algorithm to avoid poor local minima and achieve better performance. Step 4 of Algorithm 1 corresponds to this outer loop.

The inner loop (steps 5–8) does alternating minimization of θ and q in (6), for a given λ. In particular, optimization of θ (or q) for a fixed q (or θ) corresponds to solving (7) (or (8)). This alternating minimization procedure is repeated until no label switching happens. Algorithm 1 can be made more efficient by ensuring that steps 6 and 7 use the most recent θ and q as the starting points. For step 6, we employed the standard conjugate gradient method to solve (7), by converting it to an unconstrained optimization problem. For step 7, the label switching algorithm [23] was used.

The label switching algorithm assumes that the constraints are satisfied initially. It then proceeds by switching the labels of a pair of examples from two consecutive categories if the objective function decreases after such switching. The algorithm greedily performs as many such switches as possible for every consecutive categories. The pairwise switching of labels ensures that the constraints are satisfied throughout the label switching algorithm. The algorithm converges after a few iterations and the overall cost is proportional to $\mathcal{O}(n_u r)$.

Algorithm 1 SSGPOR Algorithm

1: **procedure** SSGPOR(D_l, D_u)
2: Initialize θ by solving (3).
3: Initialize the label matrix q.
4: **for** $\lambda = \{10^{-3}, 3 \times 10^{-3}, 10^{-2}, 3 \times 10^{-2}, 10^{-1}, 3 \times 10^{-1}, 1\}$ **do**
5: **repeat**
6: Estimate θ by solving the optimization problem (7) for fixed q.
7: Estimate q by solving the optimization problem (8) for fixed θ.
8: **until** q is unchanged during step 7
9: **end for**
10: **return** θ
11: **end procedure**

4 Experimental Results

We perform experiments on synthetic, benchmark and real-world data sets to compare the performance of the proposed SSGPOR approach (in the semi-supervised setting) with the EPGPOR approach. The EPGPOR approach is a supervised approach and does not use unlabeled data. We also compare the SSGPOR approach with the transductive ordinal regression (TOR) [13] approach. For brevity, we refer to these approaches as EPGPOR, SSGPOR and TOR. TOR used a transductive setting and therefore, for fair comparison, we also used SSGPOR in the transductive setting. The SSGPOR and EPGPOR approaches use the Gaussian kernel (1) in all the experiments. First, we conduct experiments on a synthetic data set to visualize the decision boundaries obtained using EPGPOR and SSGPOR. The generalization performance of the models is studied on several benchmark data sets. Finally, the effectiveness of SSGPOR is demonstrated on a real-world sentiment data set.

The generalization performance is compared using two metrics, *zero-one error* and *absolute error* [7]. 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.

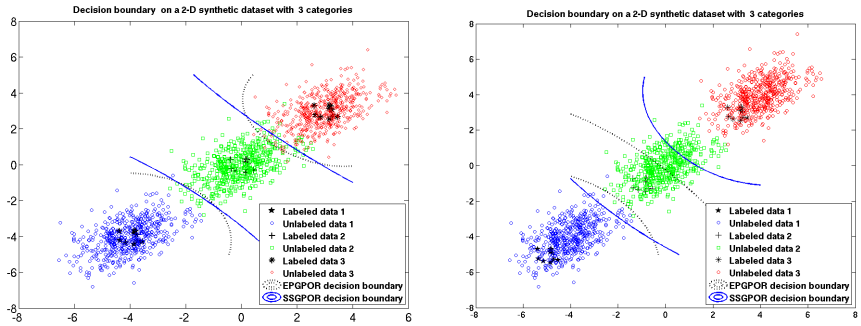
zero-one error gives the fraction of incorrect predictions on test data *i.e.* $\frac{1}{n_*} \sum_{i=1}^{n_*} \mathbb{I}(\hat{y}_i \neq y_i)$, where $\mathbb{I}(\cdot)$ is an indicator function.

absolute error gives the average deviation of predicted outputs from the actual outputs *i.e.* $\frac{1}{n_*} \sum_{i=1}^{n_*} |\hat{y}_i - y_i|$, where $|\cdot|$ denotes the absolute function.

Ordinal regression problems require the predicted category to be close enough to the actual category. The *absolute error* captures this and hence, it is more meaningful than the *zero-one error* for ordinal regression problems. One prefers approaches with low *zero-one* and *absolute* errors.

Synthetic Data We conduct experiments on a two dimensional synthetic data set to visualize the decision boundaries obtained using EPGPOR and SSGPOR. The data set consists of three ordinal categories with 10 labeled examples and 100 unlabeled examples in each category. The labeled and unlabeled data for each category were generated from a Gaussian distribution with different mean and covariance. We consider two synthetic data sets. In the first, the labeled data distribution is similar to the unlabeled data distribution while in the second, they are different. The decision boundaries obtained using SSGPOR and EPGPOR for the two data sets are depicted in Fig. 1a and Fig. 1b. The decision boundary is the predictive mean value indexed by the thresholds. Table 1 provides the zero-one and absolute errors on the unlabeled data using EPGOR and SSGPOR for both the synthetic data sets. The zero-one and absolute errors are the same in this experiment because error occurred only between the neighboring classes.

In Fig. 1a, where labeled and unlabeled data distributions are similar, both SSGPOR and EPGPOR are able to learn decision boundaries passing through a low density region. In Fig. 1b, where the labeled data distribution differs from the unlabeled data distribution, SSGPOR learns a better decision boundary



(a) Labeled data distribution similar to unlabeled data distribution

(b) Labeled data distribution not similar to unlabeled data distribution

Fig. 1. The decision boundaries obtained with SSGPOR and EPGPOR on a 2-dimensional synthetic data set with 3 ordinal categories.

Table 1. Zero-one and absolute errors on the synthetic dataset using EPGPOR and SSGPOR. The numbers in bold face style indicate the best results.

	distributions similar		distributions different	
Method	zero-one	absolute	zero-one	absolute
EPGPOR	0.0456	0.0456	0.1489	0.1489
SSGPOR	**0.0267**	**0.0267**	**0.0733**	**0.0733**

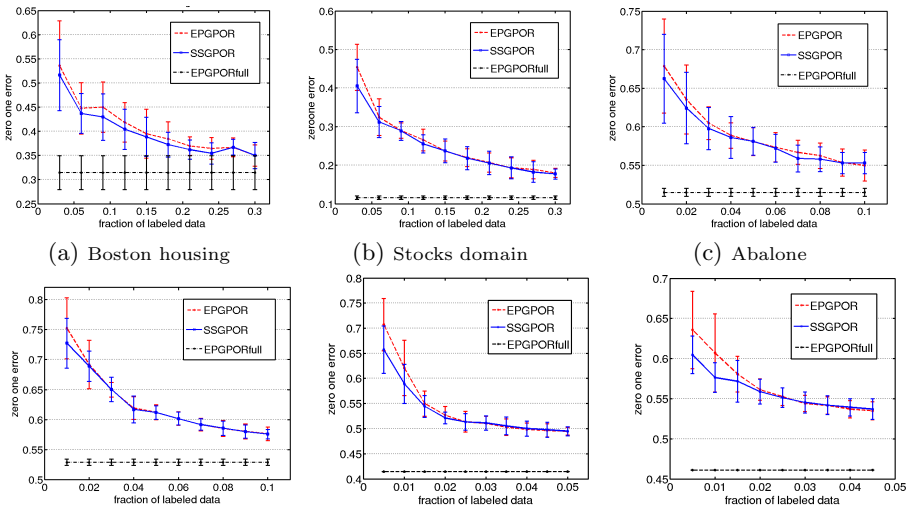
than EPGPOR. The unlabeled data help SSGPOR to shift its decision boundary towards a region of low data density. From Table 1, we observe that in either cases, SSGPOR gives lower errors than EPGPOR. It is important to note that the increase in the error is significantly higher ($\sim 10\%$) for EPGPOR compared to SSGPOR ($\sim 5\%$). This corroborates well with the observation that effective decision boundary is learnt by SSGPOR using unlabeled data.

Benchmark Data We conduct experiments on benchmark data sets to study the generalization performance of the proposed SSGPOR approach. The experiments are conducted on six benchmark data sets [7] with varying sizes. The 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. For each data set, we discretize the target values in the original data set into 5 ordinal categories. Each data set is randomly partitioned into training and test data sets as mentioned in Table 2. We generate 10 such training and test data set instances by repeated independent partitioning. For each data set, zero-one and absolute errors are obtained on all the 10 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 the approaches.

Semi-supervised Setting Figures 2 and 3 provide a comparison of SSGPOR and EPGPOR on the benchmark data sets using mean zero-one error and mean absolute error, respectively. Here, a fraction of the training data acts as

Table 2. Benchmark data sets and their properties

Data sets	Boston	Stocks	Abalone	Bank	California	Census
Attributes	13	9	8	32	8	16
Training Instances	300	600	1000	2000	3000	4000
Test Instances	206	350	3177	6192	17,640	18,784



(a) Boston housing (b) Stocks domain (c) Abalone

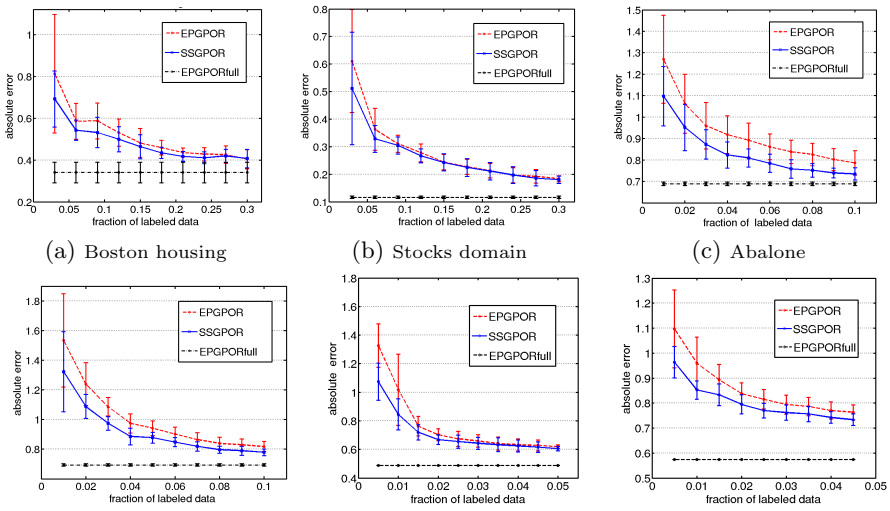
(d) Bank (e) California (f) Census

Fig. 2. Comparison of SSGPOR and EPGPOR using mean zero-one error on varying the fraction of labeled examples. Error bars denote the standard deviation.

labeled data and the rest as unlabeled data. For each benchmark data set, we plot the performance of the approaches as we vary the fraction of labeled data. We also plot the performance that can be obtained using EPGOR when the entire training set is used as the labeled data, and is denoted as EPGPORfull.

We observe from Fig. 2 and Fig. 3 that SSGPOR performs better than EPGPOR for both zero-one and absolute errors. The improvement in performance is higher when the fraction of labeled data are small. As we increase the fraction of labeled data, the improvement in performance decreases, and both the approaches start giving similar results. Eventually, the performance of both the approaches converges to the case of using full training data as the labeled data set. We observe that the improvement in performance is greater for the absolute error than for the zero-one error. That is, the labels predicted by SSGPOR are more closer to the true labels, as one would desire in an ordinal regression problem. SSGPOR gives better results on large data sets like California and Census, than on small data sets. This is due to the availability of more unlabeled data in large data sets. SSGPOR is thus able to make effective use of unlabeled data to improve the generalization performance on benchmark data sets.

Statistical Significance Test We use the paired t-test [27] to check if the proposed SSGPOR performs significantly better than EPGPOR. For each



(a) Boston housing (b) Stocks domain (c) Abalone

(d) Bank (e) California (f) Census

Fig. 3. Comparison of SSGPOR and EPGPOR using mean absolute error on varying the fraction of labeled examples. Error bars denote the standard deviation.

Table 3. T-test statistic computed with respect zero-one and absolute errors for different datasets for the smallest fraction of labeled examples. We use the bold face style to indicate the cases for which the t-test statistic is greater than the critical value.

Error	Boston	Stocks	Abalone	Bank	California	Census
Zero-one	**1.8331**	**2.9394**	1.0179	**2.2251**	**4.4971**	**2.4149**
Absolute	**2.3141**	**4.0553**	**3.4269**	**3.2525**	**4.8434**	**2.9454**

data set, we compute the t-test statistic with respect to zero-one and absolute errors for the smallest fraction of labeled data. The errors are obtained on 10 instances of training and test data sets. The null hypothesis is that both SSGPOR and EPGPOR have similar performance. Under the null hypothesis, the t-test statistic follows the Student's t-distribution with 9 degrees of freedom[1]. For the confidence level of 95% and degrees of freedom 9, critical value for the one-sided t-test is 1.833. We reject the null hypothesis if the computed t-test statistic is greater than the critical value. Table 3 reports the t-test statistic computed for each dataset. From Table 3, we observe that the computed t-statistic with respect to zero-one error is greater than the critical value for all datasets except for the Abalone data set. With respect to absolute error, it is greater than the critical value for all the data sets. Therefore, the performance of SSGPOR is significantly better than that of EPGPOR and is a better approach than EPGPOR to perform semi-supervised ordinal regression.

[1] Under null hypothesis t-statistic follows the Student's t-distribution with $s-1$ degrees of freedom, where s is the sample size

Table 4. Comparison of SSGPOR and EPGPOR in the transductive setting for different labeled data sizes. The numbers in bold face style indicate the best results.

	50 labeled examples				100 labeled examples			
	zero-one error		absolute error		zero-one error		absolute error	
Data	EPGPOR	SSGPOR	EPGPOR	SSGPOR	EPGPOR	SSGPOR	EPGPOR	SSGPOR
Boston	0.3860	**0.3816**	0.4656	**0.4498**	0.3590	**0.3538**	0.4192	**0.4039**
Stocks	0.2732	**0.2503**	0.2894	**0.2669**	0.2079	**0.1977**	0.2165	**0.2059**
Abalone	0.5764	**0.5643**	0.8834	**0.7947**	0.5453	**0.5407**	0.7781	**0.7378**
Bank	0.6626	**0.6571**	1.1657	**1.0287**	0.6130	**0.6091**	0.9358	**0.8756**
California	0.5253	**0.5141**	0.6998	**0.6649**	0.4976	**0.4934**	0.6331	**0.6201**
Census	0.5837	**0.5823**	0.9028	**0.8566**	0.5553	**0.5540**	0.8215	**0.7822**

Table 5. Comparison of EPGPOR, SSGPOR and TOR when labeled data size is 100. The numbers in bold face style indicate the best results.

	zero-one error			absolute error		
Data	EPGPOR	SSGPOR	TOR	EPGPOR	SSGPOR	TOR
Abalone	0.5453	**0.5407**	0.5420	0.7781	**0.7378**	0.7700
Bank	0.6130	**0.6091**	0.6220	0.9358	**0.8756**	0.9200
California	0.4976	**0.4934**	0.5200	0.6331	**0.6201**	0.6750
Census	0.5553	**0.5540**	0.5700	0.8215	**0.7822**	0.7900

Transductive Setting We conduct experiments to study the performance of the proposed approach in a transductive setting. Here, we assume the unlabeled test examples are available at the time of training. The experiments are conducted on all the data sets. The mean zero-one and absolute errors (over 20 independent partitions of training and test data), when labeled data sizes are 50 and 100, are given in Table 4. Transductive setting experiments show a similar behavior as that of the semi-supervised setting. Comparison with EPGPOR shows that the improvement in performance is higher when the fraction of labeled data are small and the improvement decreases with more labeled data. Again, we observe that the improvements are larger for the absolute error than for the zero-one error.

Comparison with TOR [13] The transductive setting experiments provide us an opportunity to compare EPGPOR and SSGPOR with TOR. We note that TOR uses a Perceptron kernel [13]. Table 5 compares the mean zero-one and absolute errors obtained for EPGPOR and SSGPOR with the reported TOR results [13] on Abalone, Bank, California and Census data sets, when the labeled data size is fixed to 100. We observe that the performance of EPGPOR is comparable with that of TOR whereas, SSGPOR performs better than TOR. Also, we get the predictive probability information using SSGPOR unlike TOR.

Sentiment Data We conduct experiments on real-world sentiment data sets[2]. The data sets consist of reviews and ratings of users on products at *Amazon.com* [13]. The task is to predict the rating of a user review on a scale of 1 to

[2] http://www.cs.jhu.edu/∼mdredze/datasets/sentiment/

Table 6. Mean zero one and absolute errors on the sentiment data when labeled data size is 100. The numbers in bold face style indicate the best results.

	zero-one error		absolute error	
Data	EPGPOR	SSGPOR	EPGPOR	SSGPOR
Book	0.7385	**0.6546**	1.3022	**0.9424**
Kitchen	0.7266	**0.6547**	1.2370	**0.9642**
Dvd	0.7276	**0.6476**	1.1558	**0.9288**
Electronics	0.7327	**0.6613**	1.3714	**0.9696**

5. We consider four categories of products, Book, Kitchen, Dvd and Electronics. The data sets are preprocessed and the best 1000 words are selected based on the *tf-idf* value to form the feature vector. The data sets consist of around 5000 samples. We conduct the transductive setting experiments on the data sets with the labeled data size as 100. Table 6 reports the mean zero one and mean absolute errors obtained using SSGPOR and EPGPOR for the data sets. We observe that SSGPOR significantly boosts the performance with the additional unlabeled data, on the sentiment data sets.

5 Conclusion

In this work, we proposed an approach to perform ordinal regression using Gaussian processes in a semi-supervised setting. A semi-supervised approach to ordinal regression is important as it is expensive to obtain labeled data, whereas unlabeled data are easily available. The proposed approach, semi-supervised Gaussian process ordinal regression (SSGPOR), was based on the assumption that the distribution on unlabeled data is similar to that on labeled data. The approach used an alternating optimization method to learn the model parameters and the label matrix. The label matrix was learnt efficiently using the label switching algorithm. Experimental results on synthetic, benchmark and real-world data sets showed that the SSGPOR approach performed better than the supervised EPGPOR approach and the TOR approach. Thus, it is a useful approach for semi-supervised ordinal regression.

References

1. McCullagh, P.: Regression Models for Ordinal Data. Journal of the Royal Statistical Society **42** (1980) 109–142
2. McCullagh, P., Nelder, J.A.: Generalized Linear Models (Second edition). London: Chapman & Hall (1989)
3. Johnson, V.E., Albert, J.H.: Ordinal Data Modeling (Statistics for Social and Behavioral Sciences). Springer (2001)
4. Kramer, S., Widmer, G., Pfahringer, B., De Groeve, M.: Prediction of Ordinal Classes Using Regression Trees. Fundam. Inform. **47** (2001) 1–13
5. Frank, E., Hall, M.: A simple approach to ordinal classification. In: European Conference on Machine Learning. (2001) 145–156

6. Chu, W., Keerthi, S.S.: New Approaches to Support Vector Ordinal Regression. In: International Conference on Machine Learning. (2005) 145–152
7. Chu, W., Ghahramani, Z.: Gaussian Processes for Ordinal Regression. J. Mach. Learn. Res. **6** (2005) 1019–1041
8. Chapelle, O., Schölkopf, B., Zien, A., eds.: Semi-Supervised Learning. MIT Press, Cambridge, MA (2006)
9. Herbrich, R., Graepel, T., Obermayer, K.: Large Margin Rank Boundaries for Ordinal Regression. In: Advances in Large Margin Classifiers, MIT Press (2000)
10. Shashua, A., Levin, A.: Ranking with Large Margin Principle: Two Approaches. In: Advances in Neural Information Processing Systems. (2003) 937–944
11. Li, L., Lin, H.T.: Ordinal Regression by Extended Binary Classification. In: Advances in Neural Information Processing Systems. (2006) 865–872
12. 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** (2010) 906–910
13. Seah, C.W., Tsang, I., Ong, Y.S.: Transductive Ordinal Regression. IEEE Transactions on Neural Networks and Learning Systems **23**(7) (2012) 1074–1086
14. Liu, Y., Liu, Y., Zhong, S., Chan, K.C.: Semi-Supervised Manifold Ordinal Regression for Image Ranking. In: ACM Multimedia. (2011) 1393–1396
15. Tresp, V.: A Bayesian Committee Machine. Neural Computation **12**(11) (2000)
16. Lawrence, N.D., Jordan, M.I.: Semi-supervised Learning via Gaussian Processes. In: Advances in Neural Information Processing Systems. (2004) 753–760
17. Sindhwani, V., Chu, W., Keerthi, S.S.: Semi-supervised Gaussian process classifiers. In: International Joint Conference on Artificial Intelligence. (2007) 1059–1064
18. Guo, X., Yasumura, Y., Uehara, K.: Semi-supervised gaussian process regression and its feedback design. In: Advanced Data Mining and Applications. Volume 7713 of Lecture Notes in Computer Science. (2012) 353–366
19. Adams, R.P., Ghahramani, Z.: Archipelago: Nonparametric Bayesian Semi-Supervised Learning. In: International Conference on Machine Learning. (2009)
20. Quadrianto, N., Petterson, J., Smola, A.: Distribution Matching for Transduction. In: Advances in Neural Information Processing Systems. (2009) 1500–1508
21. Le, Q.V., Smola, A.J., Gärtner, T., Altun, Y.: Transductive Gaussian Process Regression with Automatic Model Selection. In: European Conference on Machine Learning. (2006) 306–317
22. Gärtner, T., Le, Q.V., Burton, S., Smola, A.J., Vishwanathan, S.V.N.: Large-Scale Multiclass Transduction. In: Advances in Neural Information Processing Systems. (2006) 411–418
23. Keerthi, S.S., Sellamanickam, S., Shevade, S.K.: Extension of TSVM to Multi-Class and Hierarchical Text Classification Problems With General Losses. In: International Conference on Computational Linguistics. (2012) 1091–1100
24. Sindhwani, V., Keerthi, S.S., Chapelle, O.: Deterministic Annealing for Semi-Supervised Kernel Machines. In: International Conference on Machine Learning. (2006) 841–848
25. Rasmussen, C.E., Williams, C.K.I.: Gaussian Processes for Machine Learning (Adaptive Computation and Machine Learning). The MIT Press (2005)
26. Minka, T.: A Family of Algorithms for Approximate Bayesian Inference. PhD thesis, Massachusetts Institute of Technology (2001)
27. Dietterich, T.G.: Approximate Statistical Tests for Comparing Supervised Classification Learning Algorithms. Neural Computation **10** (1998) 1895–1923