Gaussian Process Learning from Order Relationships using Expectation Propagation

Ruixuan Wang and Stephen J. McKenna
School of Computing, University of Dundee
{*ruixuanwang, stephen*}*@computing.dundee.ac.uk*

Abstract

A method for Gaussian process learning of a scalar function from a set of pair-wise order relationships is presented. Expectation propagation is used to obtain an approximation to the log marginal likelihood which is optimised using an analytical expression for its gradient. Experimental results show that the proposed method performs well compared with a previous method for Gaussian process preference learning. [1]

1. Introduction

Given a collection of instances $\mathcal{X} = \{\mathbf{x}_n, n = 1, \ldots, N\}$ and a set of M noisy labels, $\mathcal{D} = \{\mathbf{v}_m \succ \mathbf{u}_m, m = 1, \ldots, M\}$, where each label indicates an order relationship (denoted $\succ$) between two of the instances, an interesting learning problem is to infer a scalar target function $f(\mathbf{x})$ that approximately satisfies the order relationships specified by the labels. This problem arises when learning users' preferences for certain products or news topics [8], learning to rank [1], and in multiclass classification [3]. In this paper, we use Gaussian Process (GP) learning [3, 4, 8]. GP can automatically determine its free parameters for model selection and has been extensively applied to classification and regression [6, 7]. Chu and Ghahramani [3] applied GP to learning preferences over pairs of instances. They used a Laplace approximation (LA) of the posterior at its maximum a posteriori (MAP) estimate and gradient-based optimization of the resulting approximate evidence. Expectation Propagation (EP) [5] with evidence maximization or variational methods has been used to learn GP models for classification [6] or ordinal regression [2]. While EP has consistently shown better performance than LA for classification [6], it is not clear how EP performs in the proposed scenario, i.e. learning from order relationships. EP has been used with a variational method [1] to maximise a lower bound on the marginal likelihood. The method proposed in this paper avoids the need for this further approximation. It uses EP to enable evidence maximisation and provides an analytical formula for the gradient of the approximate marginal likelihood. In contrast to EP for classification and regression which deals with likelihood functions which are products of functions of a single latent variable, the EP method here deals with likelihood functions in which each factor is a function of a linear combination of latent variables.

2. GP for Preference Learning

Chu and Ghahramani formulated preference learning as Gaussian process learning [1, 3]. This formulation is summarised in this section for completeness. In Gaussian process learning it is assumed that a latent function value $f(\mathbf{x}_n)$ is associated with each instance $\mathbf{x}_n$ [7]. The posterior is

$$p(\mathbf{f}|\mathcal{X}, \mathcal{D}) = \frac{p(\mathbf{f}|\mathcal{X})\, p(\mathcal{D}|\mathbf{f})}{Z}\,, \qquad (1)$$

where $\mathbf{f} = [f(\mathbf{x}_1)\ f(\mathbf{x}_2)\ \ldots\ f(\mathbf{x}_N)]^{\mathsf{T}}$, and the denominator $Z = p(\mathcal{D}|\mathcal{X}) = \int p(\mathbf{f}|\mathcal{X})\, p(\mathcal{D}|\mathbf{f}) d\mathbf{f}$ is called the evidence or marginal likelihood. An appropriate form for the likelihood function is [3]

$$\begin{aligned} p(\mathcal{D}|\mathbf{f}) &= \prod_{m=1}^{M} p(\mathbf{v}_m \succ \mathbf{u}_m | f(\mathbf{v}_m), f(\mathbf{u}_m)) \\ &= \prod_{m=1}^{M} \Phi(\frac{\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{f}_m}{\sqrt{2}\sigma})\,, \end{aligned} \qquad (2)$$

where $\mathbf{f}_m = [f(\mathbf{v}_m)\ f(\mathbf{u}_m)]^{\mathsf{T}}$, $\boldsymbol{\alpha} = [1\ \ -1]^{\mathsf{T}}$, and $\Phi(\cdot)$ is the cumulative normal distribution function. The variance term σ^2 should depend on the reliability of the labels. Its inverse $\gamma = \sigma^{-2}$ is known as the precision.

The Gaussian process prior is

$$p(\mathbf{f}|\mathcal{X}) = \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{C})\,, \tag{3}$$

where the covariance matrix $\mathbf{C}$ has entries $C(i,j) = \exp\{-\frac{1}{2\ell^2}d^2(\mathbf{x}_i, \mathbf{x}_j)\}$, and $d(\mathbf{x}_i, \mathbf{x}_j)$ is an appropriate distance measurement between $\mathbf{x}_i$ and $\mathbf{x}_j$. All experiments reported in this paper used Euclidean distance.

The hyper-parameters in this model are the variance σ^2 and the length-scale ℓ. Learning can be formulated as searching for hyperparameter values that maximize the marginal likelihood $p(\mathcal{D}|\mathcal{X})$. This marginal likelihood is analytically intractable because the likelihood (and therefore the posterior) is non-Gaussian. Chu and Ghahramani [3] used Laplace's method to approximate the posterior as a Gaussian.

3 Transformation of Marginal Likelihood

By defining $\mathbf{g} = \sigma^{-1}\mathbf{f}$ and $\mathbf{K} = \sigma^{-2}\mathbf{C}$ [6], the marginal likelihood can be written

$$\begin{aligned} Z &= \int \mathcal{N}(\mathbf{f}|\mathbf{0}, \mathbf{C}) \prod_{m=1}^{M} \Phi(\frac{\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{f}_m}{\sqrt{2}\sigma})\, d\mathbf{f} \\ &= \int \mathcal{N}(\mathbf{g}|\mathbf{0}, \mathbf{K}) \prod_{m=1}^{M} \Phi(\frac{\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{g}_m}{\sqrt{2}})\, d\mathbf{g} \\ &= \int p(\mathbf{g}|\mathcal{X})\, p(\mathcal{D}|\mathbf{g})\, d\mathbf{g}\,, \end{aligned} \tag{4}$$

where $p(\mathbf{g}|\mathcal{X}) = \mathcal{N}(\mathbf{g}|\mathbf{0}, \mathbf{K})$ and $p(\mathcal{D}|\mathbf{g}) = \prod_{m=1}^{M} \Phi(\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{g}_m/\sqrt{2})$. A motivation for this transformation is that the hyperparameters appear only in the kernel matrix $\mathbf{K}$ which has entries $K(i,j) = k(\mathbf{x}_i, \mathbf{x}_j) = \gamma \exp\{-\frac{1}{2\ell^2}d^2(\mathbf{x}_i, \mathbf{x}_j)\}$. This eases the problem of hyperparameter estimation, as shown in Section 5. The predictive distribution $p(f(\mathbf{r})|\mathbf{r}, \mathcal{X}, \mathcal{D})$ for a latent function $f(\mathbf{r})$ at any test instance $\mathbf{r}$ can be directly computed from $p(g(\mathbf{r})|\mathbf{r}, \mathcal{X}, \mathcal{D})$ as

$$p(f(\mathbf{r})|\mathbf{r}, \mathcal{X}, \mathcal{D}) \;=\; \frac{1}{\sqrt{\gamma}} p(g(\mathbf{r})|\mathbf{r}, \mathcal{X}, \mathcal{D})\,. \tag{5}$$

Furthermore, given two test instances $\mathbf{r}$ and $\mathbf{s}$, the probability of their order relationship $p(\mathbf{r} \succ \mathbf{s}|\mathcal{X}, \mathcal{D})$ can be computed from $p(\mathbf{g}_t|\mathcal{X}, \mathcal{D})$, where $\mathbf{g}_t = [g(\mathbf{r})\, g(\mathbf{s})]^{\mathsf{T}}$, since

$$\begin{aligned} p(\mathbf{r} \succ \mathbf{s}|\mathcal{X}, \mathcal{D}) &= \int p(\mathbf{r} \succ \mathbf{s}|\mathbf{f}_t, \mathcal{X}, \mathcal{D}) p(\mathbf{f}_t|\mathcal{X}, \mathcal{D})\, d\mathbf{f}_t \\ &= \int p(\mathbf{r} \succ \mathbf{s}|\mathbf{g}_t, \mathcal{X}, \mathcal{D}) p(\mathbf{g}_t|\mathcal{X}, \mathcal{D})\, d\mathbf{g}_t \end{aligned} \tag{6}$$

4 Expectation Propagation

EP is an iterative Bayesian inference method to approximate a posterior as a Gaussian so that the corresponding (approximate) marginal likelihood Z_{EP} can be analytically computed [5]. EP has been used to approximate non-Gaussian posteriors for Gaussian process classification and regression problems [6, 7]. In those settings, the likelihood is a product of functions of single latent variables. However, given pairwise order relationships, each factor in the likelihood is a function of a linear combination of two latent variables. In what follows, the use of EP is described for this setting. It is described in a way that makes clear how to extend it to likelihoods in which the factors are linear combinations of more than two latent variables. Furthermore, it is a generalisation of the classification/regression setting.

In this paper, EP is used to approximate $p(\mathbf{g}|\mathcal{X}, \mathcal{D})$ rather than $p(\mathbf{f}|\mathcal{X}, \mathcal{D})$ for reasons given in Section 3. It obtains a Gaussian approximation $q(\mathbf{g}|\mathcal{X}, \mathcal{D})$ by approximating each factor $\Phi(\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{g}_m/\sqrt{2})$ as a non-normalized Gaussian $\tilde{Z}_m \mathcal{N}(\mathbf{g}_m|\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Lambda}}_m^{-1})$:

$$\begin{aligned} q(\mathbf{g}|\mathcal{X}, \mathcal{D}) &= \frac{\mathcal{N}(\mathbf{g}|\mathbf{0}, \mathbf{K})}{Z_{EP}} \prod_{m=1}^{M} \tilde{Z}_m \mathcal{N}(\mathbf{g}_m|\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Lambda}}_m^{-1}) \\ &= \frac{T_M}{Z_{EP}} \mathcal{N}(\mathbf{g}|\mathbf{0}, \mathbf{K}) \mathcal{N}(\mathbf{g}|\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}}^{-1}) \prod_{m=1}^{M} \tilde{Z}_m \\ &= \mathcal{N}(\mathbf{g}|\boldsymbol{\mu}, \boldsymbol{\Omega})\,. \end{aligned} \tag{7}$$

where $\tilde{\boldsymbol{\mu}}_m$ is a 2-vector and $\tilde{\boldsymbol{\Lambda}}_m$ is the inverse of a 2×2 covariance matrix. The normalization factor T_M is generated when the local Gaussian approximations are combined into $\mathcal{N}(\mathbf{g}|\tilde{\boldsymbol{\mu}}, \tilde{\boldsymbol{\Lambda}}^{-1})$ as

$$\begin{aligned} T_M &= (2\pi)^{\frac{N}{2}-M} |\tilde{\boldsymbol{\Lambda}}|^{-\frac{1}{2}} \prod_{m=1}^{M} |\tilde{\boldsymbol{\Lambda}}_m|^{\frac{1}{2}} \\ &\quad \times \exp\{\frac{1}{2}\tilde{\boldsymbol{\mu}}^{\mathsf{T}}\tilde{\boldsymbol{\Lambda}}\tilde{\boldsymbol{\mu}} - \frac{1}{2}\Sigma_{m=1}^{M}\tilde{\boldsymbol{\mu}}_m^{\mathsf{T}}\tilde{\boldsymbol{\Lambda}}_m\tilde{\boldsymbol{\mu}}_m\} \end{aligned} \tag{8}$$

Also,

$$\begin{aligned} \tilde{\boldsymbol{\Lambda}} &= \Sigma_{m=1}^{M}\boldsymbol{\beta}_m\tilde{\boldsymbol{\Lambda}}_m\boldsymbol{\beta}_m^{\mathsf{T}}, & \tilde{\boldsymbol{\mu}} &= \tilde{\boldsymbol{\Lambda}}^{-1}\Sigma_{m=1}^{M}\boldsymbol{\beta}_m\tilde{\boldsymbol{\Lambda}}_m\tilde{\boldsymbol{\mu}}_m\,, \\ \boldsymbol{\Omega} &= (\mathbf{K}^{-1} + \tilde{\boldsymbol{\Lambda}})^{-1}, & \boldsymbol{\mu} &= \boldsymbol{\Omega}\Sigma_{m=1}^{M}\boldsymbol{\beta}_m\tilde{\boldsymbol{\Lambda}}_m\tilde{\boldsymbol{\mu}}_m\,, \end{aligned}$$

where $\boldsymbol{\beta}_m$ is an $N \times 2$ matrix that consists of the two basis vectors corresponding to the indices of $\mathbf{v}_m$ and

$\mathbf{u}_m$ in the data $\mathcal{X}$. The marginal likelihood Z_{EP} that approximates Z in Equation (4) is then obtained directly as the integral of this Gaussian:

$$\begin{aligned}\log(Z_{EP}|\boldsymbol{\theta}) = & -\frac{1}{2}\tilde{\boldsymbol{\mu}}^{\mathsf{T}}(\mathbf{K}+\tilde{\boldsymbol{\Lambda}}^{-1})^{-1}\tilde{\boldsymbol{\mu}} \\ & -\frac{1}{2}\log|\mathbf{K}+\tilde{\boldsymbol{\Lambda}}^{-1}| - \frac{N}{2}\log 2\pi \\ & + \log T_M + \Sigma_{m=1}^{M}\log\tilde{Z}_m \, . \end{aligned} \tag{9}$$

The parameters $\{\tilde{Z}_m, \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Lambda}}_m\}$ of $q(\mathbf{g}|\mathcal{X},\mathcal{D})$ are estimated by EP. At each iteration, EP firstly removes the effect of previously estimated Gaussian $\mathcal{N}(\mathbf{g}_m|\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Lambda}}_m^{-1})$ on the previously estimated Gaussian approximation $q(\mathbf{g}|\mathcal{X},\mathcal{D})$, resulting in an incomplete Gaussian approximation $q_{-m}(\mathbf{g}|\mathcal{X},\mathcal{D})$ to the posterior. Then, the new non-normalized Gaussian $\tilde{Z}_m\mathcal{N}(\mathbf{g}_m|\tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Lambda}}_m^{-1})$ can be obtained by approximating the combination of the in-complete Gaussian approximation $q_{-m}(\mathbf{g}|\mathcal{X},\mathcal{D})$ and the $\Phi(\boldsymbol{\alpha}^{\mathsf{T}}\mathbf{g}_m/\sqrt{2})$ as a Gaussian. The reader is referred to Rasmussen and Williams for further details of this procedure [7].

5 Evidence Maximization

The hyper-parameters $\boldsymbol{\theta}$ are estimated by maximizing the log marginal likelihood $\log(Z_{EP}|\boldsymbol{\theta})$ using a gradient-based method. The gradient of $\log(Z_{EP}|\boldsymbol{\theta})$ can be analytically computed as

$$\begin{aligned}\frac{\partial\log(Z_{EP})}{\partial\theta_j} = & \frac{1}{2}\tilde{\boldsymbol{\mu}}^{\mathsf{T}}(\mathbf{K}+\tilde{\boldsymbol{\Lambda}}^{-1})^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}(\mathbf{K}+\tilde{\boldsymbol{\Lambda}}^{-1})^{-1}\tilde{\boldsymbol{\mu}} \\ & -\frac{1}{2}\mathrm{tr}((\mathbf{K}+\tilde{\boldsymbol{\Lambda}}^{-1})^{-1}\frac{\partial\mathbf{K}}{\partial\theta_j}) \, , \end{aligned} \tag{10}$$

where $\frac{\partial\mathbf{K}}{\partial\gamma}$ has entries $\exp\{-\frac{1}{2\ell^2}d^2(\mathbf{x}_i,\mathbf{x}_j)\}$, and $\frac{\partial\mathbf{K}}{\partial\ell}$ has entries $\frac{\gamma d^2(\mathbf{x}_i,\mathbf{x}_j)}{\ell^3}\exp\{-\frac{1}{2\ell^2}d^2(\mathbf{x}_i,\mathbf{x}_j)\}$. Note that the parameters $\{\tilde{Z}_m, \tilde{\boldsymbol{\mu}}_m, \tilde{\boldsymbol{\Lambda}}_m\}$ need to be estimated at each iteration of gradient descent when estimating the hyper-parameters $\boldsymbol{\theta}$.

6 Prediction

Once the hyper-parameters have been determined, the model can be used to make predictions [7]. Given a test instance $\mathbf{r}$, the predictive distribution $p(g(\mathbf{r})|\mathcal{X},\mathcal{D},\mathbf{r})$ is a Gaussian with mean $\mu_r^* = \mathbf{k}_r^{\mathsf{T}}(\mathbf{K}+\tilde{\boldsymbol{\Lambda}}^{-1})^{-1}\tilde{\boldsymbol{\mu}}$, where $\mathbf{k}_r^{\mathsf{T}}$ is the covariance vector with entries $k_r(i) = k(\mathbf{r},\mathbf{x}_i)$. In addition as in [3], given any two instances $\mathbf{r}$ and $\mathbf{s}$, the probability of their order relationship can be computed easily as $p(\mathbf{r}\succ\mathbf{s}|\mathcal{X},\mathcal{D}) = \Phi(\frac{\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\mu}^*}{\sqrt{2+\boldsymbol{\alpha}^{\mathsf{T}}\boldsymbol{\Omega}^*\boldsymbol{\alpha}}})$, where $\boldsymbol{\mu}^* = [\mu_r^* \; \mu_s^*]^{\mathsf{T}}$, and $\boldsymbol{\Omega}^*$ is the 2×2 covariance matrix related to the two instances.

7 Empirical Evaluation

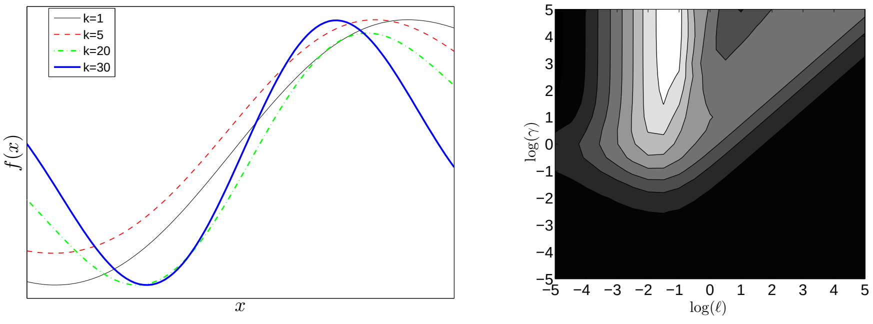
7.1 Toy Example

The method is first demonstrated on a toy problem in which data were generated using a ground-truth sine function $y = \sin(x)$. The collection $\mathcal{X}$ consisted of 50 points sampled uniformly in the interval $[-\pi,\pi)$. Each label was generated by drawing two random samples u and v from $\mathcal{X}$ and specifying $u\succ v$ if $\sin(u) > \sin(v)$, and specifying $v\succ u$ otherwise. After obtaining the approximate posterior $q(\mathbf{f}|\mathcal{X},\mathcal{D})$ (Equation 7), the mappings of 100 equally spaced points between $-\pi$ and π were computed (Section 6) and were plotted in Figure 1(a). This plot shows the results obtained when using $1, 5, 20,$ and 30 labels. As expected, the shape of the plot becomes more like that of a sine function as the number of labels increases. Note that the units of the ordinate are unimportant and the curves have been plotted so that their maxima and minima are equal. Figure 1(b) shows a plot of a log marginal likelihood $\log(Z_{EP}|\boldsymbol{\theta})$ obtained by varying the two hyper-parameters when using 30 labels. There is a maximum region around $\log(\ell) = -1.2$ and $\log(\gamma)\geq 2.0$.

7.2 Benchmark datasets

Table 1 shows a comparison with Chu and Ghahramani's Laplacian method (GPLA) on four benchmark datasets used in their paper [3]. Since the same experimental setting was used, the test results of Chu and Ghahramani are reported directly. Each dataset contained a number of multi-dimensional instances ('d' denotes the dimensionality) with corresponding scalar ground-truth output. Data were normalized so that each dimension had zero mean and unit variance. For each dataset, a specified number ('m' in Table 1) of instance pairs were randomly selected, and a label for each pair generated by comparing the ground-truth outputs for the two instances. Maximally $20,000$ pairs were randomly generated for testing. The ordering for each test pair was computed using the method (Section 6) and compared with the ground-truth ordering. The training and testing process was repeated 20 times independently. Two hyperparameter initialisations were tried: $\log\gamma = 0.0$ and $\log\ell = \log d/2$ as in [3] ('GPEP' in Table 1), and $\log\gamma = 1.0$ and $\log\ell = \log d/2$ ('GPEP2' in Table 1). In both cases, the proposed method gave significantly better test results on the Machine CPU and Boston Housing datasets, and comparable results on the other two datasets. GPEP2 generated slightly better results than GPEP.

Figure 2 illustrates the change in test error as the



(a)

(b)

Figure 1. (a) Predictions obtained using samples from a sine function and k pair-wise labels. (b) A log marginal likelihood for $k = 30$.

Table 1. Test for preference learning.

			Error Rate (%)		
Dataset	m	d	GPLA	GPEP	GPEP2
Pyrimidines	100	27	14.43 ± 2.02	13.80 ± 1.28	12.70 ± 1.08
Triazines	300	60	17.78 ± 0.97	17.12 ± 1.10	17.06 ± 0.74
MachineCpu	500	6	12.12 ± 1.49	10.16 ± 0.61	9.26 ± 0.36
BostonHouse	700	13	12.85 ± 0.46	10.28 ± 0.49	9.29 ± 0.38

number of labels is varied for the BostonHouse datset, using GPEP2 with training pairs excluded from the testing. Both the mean and standard deviation (dotted curve and vertical bars in Figure 2) of test errors decrease to convergent values ($0.088 \pm 0.003 \times 100\%$) when the number of preference pairs increases to 1000.

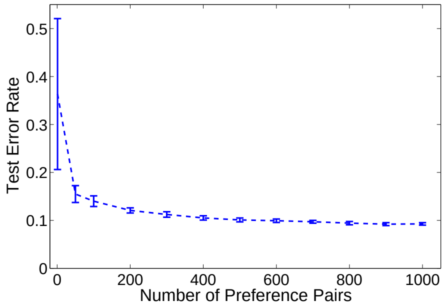


Figure 2. Change of test error.

8 Conclusion

A method for learning a Gaussian process model from data labelled with order relationships was presented. It used an analytical expression for the gradient of an approximate log marginal likelihood obtained using EP. Experimental results on benchmark datasets showed that the method performed better on some of them, and at least as well on the others, as a previous method using Laplace approximation. This suggests that the proposed method resulted in improved model selection (i.e. better estimates of the hyper-parameters) and thus improved predictions. Currently we are exploring how to actively choose pairs for labelling.

Acknowledgement

This research was supported by the UK Technology Strategy Board grant *FABRIC*, a collaborative project with Liberty Fabric, System Simulation, Calico Jack Ltd., and the Victoria and Albert Museum.

References

[1] W. Chu and Z. Ghahramani. Extensions of Gaussian processes for ranking: semi-supervised and active learning. In *Workshop Learning to Rank at NIPS 2005*, 2005.

[2] W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.

[3] W. Chu and Z. Ghahramani. Preference learning with Gaussian processes. In *Proc. of International Conference on Machine Learning*, pages 137–144, 2005.

[4] J. Fürnkranz and E. Hüllermeier. Preference learning. *Künstliche Intelligenz*, 19(1):60–61, 2005.

[5] T. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, Massachusetts Institute of Technology, 2001.

[6] H. Nickisch and C. E. Rasmussen. Approximations for binary Gaussian process classification. *Journal of Machine Learning Research*, 9:2035–2078, 2008.

[7] C. E. Rasmussen and C. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.

[8] F. Wang, B. Zhang, T. Li, W. Yin, J. Dong, and T. Li. Preference learning with extreme examples. In *Proc. of IInternational Joint Conference on Artificial Intelligence*, pages 1285–1290, 2009.