

Learning Preferences with Hidden Common Cause Relations

Kristian Kersting and Zhao Xu

Fraunhofer IAIS, Schloss Birlinghoven, 53754 Sankt Augustin, Germany
{kristian.kersting,zhao.xu}@iais.fraunhofer.de

Abstract. Gaussian processes have successfully been used to learn preferences among entities as they provide nonparametric Bayesian approaches for model selection and probabilistic inference. For many entities encountered in real-world applications, however, there are complex relations between them. In this paper, we present a preference model which incorporates information on relations among entities. Specifically, we propose a probabilistic relational kernel model for preference learning based on Silva et al.’s mixed graph Gaussian processes: a new prior distribution, enhanced with relational graph kernels, is proposed to capture the correlations between preferences. Empirical analysis on the LETOR datasets demonstrates that relational information can improve the performance of preference learning.

1 Introduction

Largely motivated by applications in search engines, information retrieval, and collaborative filtering, preference learning has recently received a lot of attention in the machine learning and information retrieval communities, see e.g. [8, 2, 1, 3, 12]. In a typical formulation, the goal of preference learning is to compare two entities such as documents, webpages, products, songs etc., and to decide which one is better or preferred e.g. by a costumer according to some application-specific criteria.

Consider a typical interaction between an information retrieval system and a user. When a user submits a query to the system, the search engine returns a list of document hyperlinks to the user, along with a title and query-related snippet extracted from the document. The user reads the list, and based on titles, snippet, and probably abstracts, decides whether a document in the list is more relevant to the query than another ones. Hence, in contrast to standard supervised learning problems such as regression and classification, preference learning is characterized by the fact that the training set consists of pairwise rankings between entities, instead of explicit entity-wise values. For example, we may only know that a webpage e_i is more relevant than another one e_j , denoted as $e_i \succ e_j$, or that a user prefers an item to another, but we do not know the exact degrees of relevance of webpages or preferences of users.

Because of their flexible nonparametric nature and good performance on regression/classification problems, Gaussian process (GP) models have recently

been explored for learning to rank [4, 5, 10]. Basically, GP based preference learning models introduce for each entity a latent variable, which is a function value $f(x_i)$ (shortened as f_i in the rest of the paper) of entity attributes x_i . We can intuitively view the latent function values as preference/relevance degrees (called preference degrees later) of entities. Then entities are ranked according to the latent values. Namely if an entity e_i is ranked above another one e_j , i.e., $e_i \succ e_j$, then the latent function value f_i of the entity is larger than that f_j of another one, i.e., $f_i > f_j$.

Existing GP ranking models, however, only exploit the available information about entity attributes and typically ignore any relations among the entities. Intuitively, however, we would like to use our information about one entity to help us reach conclusions about other, related entities. Reconsider our information retrieval example. Here, we should be able to propagate our preference relation among two documents to documents that the two documents have links to and to documents that link to the two documents.

The main contribution of the present paper is the first nonparametric Bayesian approach to learn preferences from relational data based on Gaussian processes. Specifically, we employ the concept of hidden common causes to incorporate relational information. Hidden common causes were first introduced by Silva *et al.* [21] within the *mixed graph Gaussian process* framework (XPGs) and have been demonstrated to be quite successful for classification problems. The key insight for preference learning is that some hidden, but existing common causes lurk in relational graphs, and the hidden common causes are important factors to influence the preference degrees of entities. The overall preference degree of an entity is a comprehensive result of both the entity attributes and the hidden common causes. Technically, under the GP framework, we introduce for each entity an additional latent function value $g(r_i)$ (shortened as g_i) for the relations r_i the entity participate. This latent function value encodes the preference causes hidden in the relation. Then, we model the entity preference degree ξ_i as a linear combination of related function values, i.e., f_i and g_i . In turn, each preference $e_i \succ e_j$ is modeled as a random variable conditioned on an indicator that is a function of the preference degrees ξ_i and ξ_j of the involved entities. In other words, our relational GP framework for preference learning, which we call *mixed graph preference Gaussian process* (XPGP), ranks entities taking all available information into account, attributes and relations. As shown in our experimental results on real-world LETOR datasets, a significant improvement on preference prediction quality can be achieved when employing relational information.

Our second contribution is an active exploration scheme to relational preference learning. Providing preference labels is typically quite costly as the user has e.g. to read and understand abstracts of documents. Fortunately, the uncertainty model provided by the XPGP framework, offers predictive uncertainty estimates for preferences, and therefore *naturally* – in contrast to other kernel approaches such as SVMs – allows us to develop an active exploration scheme that guides the user by asking actively for her preferences among entities so as to provide more useful observations. As our experimental analysis on a real-

world LETOR dataset showed, this improves the prediction quality faster than collecting preferences naively.

The rest of the paper is organized as follows. We start off by touching upon related work. Then, in Sec. 3, we will introduce XPGPs. Sec. 4 will develop approximate inference and learning methods, and Sec. 5 the active exploration scheme. Before concluding, we will present our experimental analysis and discuss extensions of XPGPs to domains with multiple types of relations.

2 Related Work

The present work joins two lines of research within the Gaussian process community, namely preference and relational learning.

In the first stream, Chu and Ghahramani [4] introduced a probabilistic kernel approach to ordinal regression based on Gaussian process models. In contrast to the setting discussed in this paper, the work focused on scenarios where labels of entities are ordered. Chu and Ghahramani presented a threshold model to encode the label-wise ordinal information. The work was later extended by Chu and Ghahramani to entity ranking problem (i.e. the setting discussed here) by introducing a novel likelihood function to express the entity-wise ordinal information [5]. Guiver and Snelson [10] recently presented a sparse Gaussian process model for soft ranking problem for large-scale datasets. All these models are reported to provide good performance on real-world datasets but they do not consider relational information.

The second line of research aims at incorporating relations into probabilistic kernel models. There are essentially two strategies to accomplish this. One is encoding relations in the covariance matrixes [25, 21]. The other is encoding relations as random variables conditioned on the latent function values of entities involved in relations [6, 24]. Recently Xu *et al.* [23] introduced a combination of both approach for multi-relational learning with Gaussian processes. The approach of representing relational information as hidden common causes developed by Silva *et al.* [21] is a straightforward way to encode relations, which was successfully applied on entity classification problems. So far, however, preference learning has not been considered for any of them.

Outside the Gaussian process community, several ranking and preference learning approaches have been proposed, see e.g. [12] for a nice classification of the existing approaches. Relational approaches have also been developed. For instance, Geerts *et al.* [9] provide a general ranking framework for relational databases. Agarwal [1] introduced a kernel-based approach with graph techniques (spectral relaxation), where entities and relations are respectively viewed as vertexes and edges in graphs. So the task of ranking entities is transformed to ranking vertices. It is not clear how to exploit attributes and relations simultaneously and how to distinguish different types of relations. The work closest to our is that of Qin *et al.* [18]. They proposed a kernel-based but *not probabilistic* method to rank relational entities. Their method enhanced attribute-based ranking function with the regularized Laplacian of the relational graph, then applied

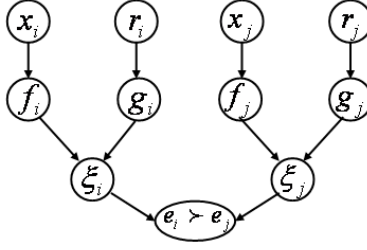


Fig. 1. Graphical representation of the XPGP model. Both entity attributes and relations are taken into account for predicting preferences $e_i \succ e_j$. Specifically, the f_i s are latent function values of entity attributes following a Gaussian process (GP) prior. Respectively, the g_i s are a latent function values of relations following another GP prior. The ξ_i s are the overall preference degree of entities. They are weighted sums of the corresponding f_i and g_i .

SVM techniques to solve the learning, i.e., optimization task. All of these approaches do not provide natural probabilistic models so that active exploration is more complicated than in our model, see e.g. [19].

3 The Model

In this section, we will introduce the XPGP model for learning preferences. Assume that there are (1) a set of n entities $E = \{e_1, \dots, e_n\}$ with attributes $X = \{x_i : x_i \in \mathbb{R}^D, i = 1, \dots, n\}$, (2) relations $R = \{r_{i,j} : i, j \in 1, \dots, n\}$ among the entities, and (3) a set of m observed pairwise preferences (a.k.a. ordinal relations/ranks) among entities, $O = \{e_{i_s} \succ e_{j_s} : s = 1, \dots, m; i_s, j_s \in 1, \dots, n\}$ (i_s and j_s are entities involved in s -th observed preference). With r_i , we will denote all relations in which entity e_i participates.

The XPGP model is graphically summarized in Fig. 1. Essentially, we introduce for each entity two latent function values $f(x_i)$ and $g(r_i)$ (shortened as f_i , g_i) such that $f(\cdot)$ and $g(\cdot)$ are functions of attributes and relations, respectively. Now, we form the linear combination of both values, i.e., $\xi_i = \omega_1 f_i + \omega_2 g_i$. The value ξ_i represents the preference degree of the entity e_i taking both attribute- and relation-wise factors into account. Finally, a preference $e_i \succ e_j$ is viewed as a random variable conditioned on the corresponding indicators of the involved entities with a likelihood distribution $P(e_i \succ e_j | \xi_i, \xi_j)$. In the following subsections, we will provide more details.

3.1 Prior Distributions

Let us start with defining the prior distributions. We essentially define priors for the attribute-wise and for the relation-wise latent function values separately and combine them using a linear model. Specifically, we assume an infinite number of latent function values $\{f_1, f_2, \dots\}$ that follow a Gaussian process prior with mean

function $m_a(x_i)$ and covariance function $k_a(x_i, x_j)$. Here we used the subscript a to emphasize that they are attribute-wise. In turn, any finite set of function values $\{f_i : i = 1, \dots, n\}$ has a multivariate Gaussian distribution with mean and covariance matrix defined in terms of the mean and covariance functions of the GP [20]. Without loss of generality, we assume zero mean so that the GP is completely specified by the covariance function only. A typical choice is the squared exponential covariance function with isotropic distance measure:

$$k_a(x_i, x_j) = \kappa^2 \exp\left(-\frac{\rho^2}{2} \sum_d^D (x_{i,d} - x_{j,d})^2\right), \quad (1)$$

where κ and ρ are parameters of the covariance function, and $x_{i,d}$ denotes the d -th dimension of the attribute vector x_i .

Similarly, we place a zero-mean GP over $\{g_1, g_2, \dots\}$. Again, $\{g_i : i = 1, \dots, n\}$ follow a multivariate Gaussian distribution. In contrast to the attribute-wise GPs, however, the covariance function $k_r(r_i, r_j)$ should represent correlation of i and j on relations. There are essentially two strategies to define such kernel functions. The simplest way is to represent the known relations of entity i as a vector. The kernel function $k_r(r_i, r_j)$ can then be any Mercer kernel function, and the computations are essentially the same as for the attributes. Alternatively, we notice that entities and relations form a graph, and we can naturally employ graph-based kernels to obtain the covariances, see e.g. [22, 25, 21]. The simplest graph kernel might be the regularized Laplacian

$$K_r = [\beta(\Delta + I/\iota^2)]^{-1}, \quad (2)$$

where β and ι are two parameters of the graph kernel. Δ denotes the combinatorial Laplacian, which is computed as $\Delta = D - W$, where W denotes the adjacency matrix of a weighted, undirected graph, i.e., $W_{i,j}$ is taken to be the weight associated with the edge between i and j encoding for examples the extent of interactions between two genes or the communication frequency between two persons. D is a diagonal matrix with entries $d_{i,i} = \sum_j w_{i,j}$.

Finally, the prior distributions of f and g are combined as follows:

$$P(f, g | X, R) = \frac{1}{(2\pi)^n |K_a|^{\frac{1}{2}} |K_r|^{\frac{1}{2}}} \exp\left(-\frac{f^T K_a^{-1} f + g^T K_r^{-1} g}{2}\right),$$

where f and g denote $\{f_1, \dots, f_n\}$ and $\{g_1, \dots, g_n\}$. K_a (resp. K_r) denotes the $n \times n$ covariance matrix whose ij -th entry is computed with the corresponding covariance function.

3.2 Preference Likelihood

What is left is the definition of the preference likelihood. We essentially extend Chu and Ghahramani's likelihood function to the relational case [5]. Recall that, in the relational case, the preference degree of an entity consists of two components: the attribute-wise factor and the relation-wise factor, respectively

represented as the latent function values f_i and g_i . To combine both we represent the overall preference degree of the entity as the weighted sum of both latent functions, i.e.,

$$\xi_i = \omega_1 f_i + \omega_2 g_i.$$

In the ideal, noise-free case it is natural to assume that *if e_i is preferred to e_j , then the preference degree of e_i is larger than that of e_j* . Denoting the non-contaminated preference degrees as $\tilde{\xi}_i$ and $\tilde{\xi}_j$, we have

$$P(e_i \succ e_j \mid \tilde{\xi}_i - \tilde{\xi}_j) = \begin{cases} 1 & \text{if } \tilde{\xi}_i - \tilde{\xi}_j \geq 0 \\ 0 & \text{otherwise.} \end{cases}$$

For real-world situations, however, it is more realistic to consider latent function values corrupted by Gaussian noise, i.e., $\xi_i = \tilde{\xi}_i + \epsilon$, $\epsilon \sim N(0, \sigma^2)$ and $P(\tilde{\xi}_i - \tilde{\xi}_j \mid \xi_i - \xi_j) = N(\cdot \mid \xi_i - \xi_j, 2\sigma^2)$. Now, we can define the preference likelihood function $P(e_i \succ e_j \mid \xi_i - \xi_j)$ as follows

$$\int P(e_i \succ e_j \mid \tilde{\xi}_i - \tilde{\xi}_j) P(\tilde{\xi}_i - \tilde{\xi}_j \mid \xi_i - \xi_j) d(\tilde{\xi}_i - \tilde{\xi}_j) = \int_{-\infty}^{\frac{\xi_i - \xi_j}{\sqrt{2}\sigma}} N(t \mid 0, 1) dt \equiv \Phi\left(\frac{\xi_i - \xi_j}{\sqrt{2}\sigma}\right).$$

This encodes the natural assumption:

The larger the difference between the preference degrees of e_i and e_j , the more likely is it that e_i is preferred to e_j .

Finally, the marginal likelihood $P(O \mid \xi)$ of m observed ordinal relations given preference degrees $\xi = \{\xi_1, \dots, \xi_n\}$ can be found to be

$$\prod_s^m P(e_{i_s} \succ e_{j_s} \mid \xi_{i_s}, \xi_{j_s}) = \prod_s^m \Phi\left(\frac{\xi_{i_s} - \xi_{j_s}}{\sqrt{2}\sigma}\right).$$

4 EMEP-based Approximate Inference and Learning

So far we have described the XPGP model. In this section, we will present approximate algorithms for inferring posterior distributions, for making preference predictions, and for estimating the hyperparameters.

4.1 Posterior Inference

The key inference problem is computing the posterior distribution of the latent function values given attributes X , relations R , and preferences O , i.e.,

$$P(f, g \mid X, R, O) \propto P(f, g \mid X, R) \prod_s^m P(e_{i_s} \succ e_{j_s} \mid f_{i_s}, g_{i_s}, f_{j_s}, g_{j_s}). \quad (3)$$

Unfortunately, computing the posterior distribution is intractable for two reasons: (1) $P(e_{i_s} \succ e_{j_s} \mid f_{i_s}, g_{i_s}, f_{j_s}, g_{j_s})$ is not conjugated to the Gaussian prior

distribution and (2) the attribute-wise GP and the relation-wise GP are coupled together according to the overall preference degrees that are weighted sums of attribute-wise and relation-wise latent function values. Therefore, we stick to the expectation propagation (EP) algorithm [17] to approximate the posterior distribution, which we will now derive.

First, to counterattack the computation complexity due to (2), i.e., due to the coupling of the different GPs, we introduce a new variable

$$\hat{\xi}_i = \frac{\omega_1}{\sqrt{2}\sigma} f_i + \frac{\omega_2}{\sqrt{2}\sigma} g_i. \quad (4)$$

Since $f = \{f_1, \dots, f_n\}$ and $g = \{g_1, \dots, g_n\}$ respectively follow two independent multivariate Gaussian distributions with mean zero and covariance matrixes K_a and K_r , we have that $\hat{\xi} = \{\hat{\xi}_1, \dots, \hat{\xi}_n\}$ also follows a multivariate Gaussian distribution with mean zero and covariance matrix:

$$K = \frac{\omega_1^2}{2\sigma^2} K_a + \frac{\omega_2^2}{2\sigma^2} K_r. \quad (5)$$

Now, we convert the computation of the posterior distribution of f and g in Eq. (3) to the computation of the posterior distribution of $\hat{\xi}$:

$$P(\hat{\xi}|X, R, O) \propto P(\hat{\xi}|X, R) \prod_{s=1}^m P(e_{i_s} \succ e_{j_s} | \hat{\xi}_{i_s}, \hat{\xi}_{j_s}), \quad (6)$$

where the prior is a Gaussian distribution with covariance matrix as defined in Eq. (5). The likelihood function is $\Phi(\hat{\xi}_{i_s} - \hat{\xi}_{j_s})$, where $\Phi(\cdot)$ is the cumulative Gaussian distribution. Without loss of generality, we assume its mean and variance are zero and one respectively.

Now, we tackle the computational complexity due to (1), i.e., due to the non-conjugation of the likelihood and the prior distributions. In the EP framework, we use unnormalized Gaussian distributions

$$t_s(\hat{\xi}_{i_s}, \hat{\xi}_{j_s} | \tilde{\mu}_s, \tilde{\Sigma}_s, \tilde{Z}_s) \equiv \tilde{Z}_s \mathcal{N}(\hat{\xi}_{i_s}, \hat{\xi}_{j_s} | \tilde{\mu}_s, \tilde{\Sigma}_s) \quad (7)$$

to approximate the real likelihood distributions $\Phi(\hat{\xi}_{i_s} - \hat{\xi}_{j_s})$ where \tilde{Z}_s is a real-valued scale, the unnormalized term. Since each t_s is a Gaussian distribution, the approximate posterior $q(\hat{\xi}) = \mathcal{N}(\hat{\xi} | \mu, \Sigma) \approx P(\hat{\xi} | X, R, O)$ is also a Gaussian distribution, whose mean μ and covariance matrix Σ can be found to be

$$\begin{aligned} \Sigma &= (K^{-1} + \tilde{\Sigma}_1^{-1} + \dots + \tilde{\Sigma}_m^{-1})^{-1} \\ \text{and } \mu &= \Sigma(\tilde{\Sigma}_1^{-1} \tilde{\mu}_1 + \dots + \tilde{\Sigma}_m^{-1} \tilde{\mu}_m). \end{aligned} \quad (8)$$

Note that the t_s are two-dimensional, but Σ and μ are $n \times n$ and n dimensions, thus we need to extend $\tilde{\Sigma}_s^{-1}$ to a $n \times n$ matrix and $\tilde{\mu}_s$ to a n -dimensional column vector by adding zeros in the corresponding positions in the computation.

The approximate distributions are sequentially updated until convergence. Specifically, the approximate distribution t_s is updated at iteration $t+1$ so that it satisfies:

$$Q_{-s} \times P(e_{i_s} \succ e_{j_s} | \hat{\xi}_{i_s}, \hat{\xi}_{j_s}) \leftarrow Q_{-s} \times t_s(\hat{\xi}_{i_s}, \hat{\xi}_{j_s} | \tilde{\mu}_s^{(t+1)}, \tilde{\Sigma}_s^{(t+1)}, \tilde{Z}_s^{(t+1)}),$$

$$\text{where } Q_{-s} = \frac{\mathcal{N}(\hat{\xi}_{i_s}, \hat{\xi}_{j_s} | \mu_s^{(t)}, \Sigma_s^{(t)})}{t_s(\hat{\xi}_{i_s}, \hat{\xi}_{j_s} | \tilde{\mu}_s^{(t)}, \tilde{\Sigma}_s^{(t)}, \tilde{Z}_s^{(t)})}. \quad (9)$$

Both sides of Eq. (9) are the approximate marginal distributions of $\hat{\xi}_{i_s}$ and $\hat{\xi}_{j_s}$. They are the integrals of the approximations to Eq. (6): the left side replaces all actual likelihood distributions with approximations from the t -th iteration except for the likelihood of the preference pair currently being updated; the right side also replaces all actual likelihood distributions with approximations, but the approximation to the likelihood of the preference pair being updated is a new one, i.e., the mean, covariance matrix and unnormalized term need to be updated now. To satisfy Eq. (9), we only need to match their first and second moments [20]. This can be found to yield the following equations for computing $\tilde{\mu}_s^{(t+1)}$, $\tilde{\Sigma}_s^{(t+1)}$ and $\tilde{Z}_s^{(t+1)}$:

$$\Sigma_{-s} = (\Sigma_s^{-1} - \tilde{\Sigma}_s^{-1})^{-1}; \quad \mu_{-s} = \Sigma_{-s}(\Sigma_s^{-1}\mu_s - \tilde{\Sigma}_s^{-1}\tilde{\mu}_s). \quad (10)$$

$$z = \frac{\varrho^T \mu_{-s}}{\sqrt{1 + \varrho^T \Sigma_{-s} \varrho}}; \quad Z = \Phi(z); \quad S = \frac{1}{\sqrt{2\pi}} \exp(-\frac{z^2}{2}); \quad \varrho^T = [1, -1];$$

$$\hat{\mu}_s = \mu_{-s} + \frac{y_s S \Sigma_{-s} \varrho}{Z \sqrt{1 + \varrho^T \Sigma_{-s} \varrho}}; \quad \hat{\Sigma}_s = \Sigma_{-s} - \frac{z S Z + S^2}{Z^2 (1 + \varrho^T \Sigma_{-s} \varrho)} \Sigma_{-s} \varrho \varrho^T \Sigma_{-s} \quad (11)$$

$$\tilde{\Sigma}_s^{(t+1)} = (\hat{\Sigma}_s^{-1} - \Sigma_{-s}^{-1})^{-1}; \quad \tilde{\mu}_s^{(t+1)} = \tilde{\Sigma}_s^{(t+1)}(\hat{\Sigma}_s^{-1}\hat{\mu}_s - \Sigma_{-s}^{-1}\mu_{-s}); \quad \tilde{Z}_s^{(t+1)} = CZ;$$

$$C = (2\pi)^{-1} |\Sigma_{-s} + \tilde{\Sigma}_s|^{\frac{1}{2}} \exp(\frac{1}{2}(\mu_{-s} - \tilde{\mu}_s)^T (\Sigma_{-s} + \tilde{\Sigma}_s)^{-1} (\mu_{-s} - \tilde{\mu}_s)) \quad (12)$$

Eq. (10) is computing mean and covariance matrix of Q_{-s} . $\tilde{\mu}_s$ and $\tilde{\Sigma}_s$ are the mean and covariance matrix of the approximate distribution t_s , optimized in the last iteration t . To avoid a cluttering of notation, we do not use the superscript (t) to highlight the iteration. μ_s and Σ_s are the mean and the covariance matrix of the approximate posterior distribution of $\hat{\xi}_{i_s}$ and $\hat{\xi}_{j_s}$ that is the marginalized $q(\hat{\xi})$. Eq. (11) is computing the mean and the covariance matrix of the distribution at the left side of Eq. (9), which is derived by moment matching [20]. Finally, y_s is one if e_{i_s} is preferred to e_{j_s} , i.e. $e_{i_s} \succ e_{j_s}$; -1 otherwise.

At convergence, we obtain the optimized EP parameters that can be used to compute the approximate posterior distribution of $\hat{\xi}$ using Eq. (8).

4.2 Transductive Preference Prediction

The key prediction problem in preference learning is to predict preferences of new pairs of entities. Here, we consider the predictive inference in a transductive

setting, i.e. there is no new entity introduced in prediction. Once the procedure of posterior inference reaches stationarity, we obtain the optimized distribution on $\hat{\xi} = \{\hat{\xi}_1, \dots, \hat{\xi}_n\}$ with the EP parameters $\{\tilde{\mu}_s, \tilde{\Sigma}_s, \tilde{Z}_s\}_{s=1}^m$. It can be used to approximate the predictive distribution of the preference pair s' on entities i' and j' as follows

$$\begin{aligned} P(e_{i'} \succ e_{j'} | X, R, O) &= \int P(e_{i'} \succ e_{j'} | \hat{\xi}_{i'}, \hat{\xi}_{j'}) P(\hat{\xi}_{i'}, \hat{\xi}_{j'} | X, R, O) d\hat{\xi}_{i'} d\hat{\xi}_{j'} \\ &\approx \int \Phi(\hat{\xi}_{i'} - \hat{\xi}_{j'}) \mathcal{N}(\hat{\xi}_{i'}, \hat{\xi}_{j'} | \mu_{s'}, \Sigma_{s'}) d\hat{\xi}_{i'} d\hat{\xi}_{j'}, \\ &= \Phi\left(\frac{\varrho^T \mu_{s'}}{\sqrt{1 + \varrho^T \Sigma_{s'} \varrho}}\right), \end{aligned}$$

where $\mathcal{N}(\hat{\xi}_{i'}, \hat{\xi}_{j'} | \mu_{s'}, \Sigma_{s'})$ is the marginalized $q(\hat{\xi})$. It is the approximation to the real marginal posterior distribution $P(\hat{\xi}_{i'}, \hat{\xi}_{j'} | X, R, O)$. Since $q(\hat{\xi})$ is Gaussian, so the approximation is still Gaussian, and its mean $\mu_{s'}$ and covariance matrix $\Sigma_{s'}$ are the corresponding entries of μ and Σ (Eq. 8). The preference relation $e_{i'} \succ e_{j'}$ is conditioned on their difference on preference degree, i.e. $\hat{\xi}_{i'} - \hat{\xi}_{j'}$. Since both $\hat{\xi}_{i'}$ and $\hat{\xi}_{j'}$ are Gaussian random variables, their difference is Gaussian, too, with mean and variance $\varrho^T \mu_{s'}$ respectively $\varrho^T \Sigma_{s'} \varrho$, where ϱ denotes a column vector $[1, -1]^T$. Thus we have $\hat{\xi}_{i'} - \hat{\xi}_{j'} \sim \mathcal{N}(\cdot | \varrho^T \mu_{s'}, \varrho^T \Sigma_{s'} \varrho)$, where

$$\varrho^T \mu_{s'} \quad (13)$$

is just the difference of the means of the two preference degrees ($\hat{\xi}_{i'}$ and $\hat{\xi}_{j'}$). The variance $\varrho^T \Sigma_{s'} \varrho$ just equals

$$\text{Var}(\hat{\xi}_{i'}) + \text{Var}(\hat{\xi}_{j'}) - 2\text{Cov}(\hat{\xi}_{i'}, \hat{\xi}_{j'}) . \quad (14)$$

The larger the variance, the more uncertain we are about the preference relation.

4.3 Hyperparameter Estimation

Finally, we will describe how to estimate the hyperparameters under the empirical Bayesian framework. The hyperparameters of the XPGP model consists of the parameters of the kernel functions as well as the mixing weights of the attribute-wise and the relation-wise GPs. Note, however, that the mixing weights are just scaling the latent function values. Therefore, we can directly integrate them into the covariance functions. In other words, estimating the hyperparameters of the XPGP model can be reduced to estimating the hyperparameters of the covariance functions.

Let us denote all hyperparameters as θ . We now need to seek θ^* that maximizes the log-likelihood of the data, i.e.

$$\theta^* = \arg \max_{\theta} \log P(O | \theta, X, R) = \arg \max_{\theta} \log \int P(\hat{\xi} | \theta, X, R) P(O | \hat{\xi}) d\hat{\xi},$$

where the prior $P(\hat{\xi} | \theta, X, R)$ is a Gaussian distribution. Unfortunately, the likelihood $P(O | \hat{\xi})$ is not Gaussian, thus the integral is analytically intractable.

To solve the problem, we follow an approximate Expectation Maximization (EM) approach [14]. The algorithm alternates the following steps until convergence. In the **E-step**, given the hyperparameters, the EP parameters $(\tilde{\mu}_s, \tilde{\Sigma}_s, \tilde{Z}_s)$ are optimized to approximate the posterior distribution of the latent function variables with the current values of hyperparameters. In the **M-step**, given the density of the preference degrees, the hyperparameters are selected to maximize a lower bound of the marginal likelihood:

$$\int q(\hat{\xi}) \log \frac{P(O|\hat{\xi})P(\hat{\xi}|\theta, X, R)}{q(\hat{\xi})} d\hat{\xi} = \int q(\hat{\xi}) \log P(\hat{\xi}|\theta, X, R) d\hat{\xi} + \int q(\hat{\xi}) \log P(O|\hat{\xi}) d\hat{\xi} - \int q(\hat{\xi}) \log q(\hat{\xi}) d\hat{\xi}. \quad (15)$$

Note that the last two terms on the right-hand side are independent of the hyperparameters θ , thus we only need to optimize the first term, i.e., $\mathcal{L} :=$

$$\int q(\hat{\xi}) \log P(\hat{\xi}|\theta, X, R) d\hat{\xi} = -\frac{1}{2} \log |2\pi K| - \frac{1}{2} \mathbb{E}_q[\hat{\xi}]^T K^{-1} \mathbb{E}_q[\hat{\xi}] - \frac{1}{2} \text{tr}(K^{-1} \Sigma),$$

where K is the covariance matrix of the prior of $\hat{\xi}$ as defined in Eq. (5), and $\mathbb{E}_q[\hat{\xi}]$ is the expectation of $\hat{\xi}$ on the approximate posterior $q(\hat{\xi})$, i.e. the mean μ of $q(\hat{\xi})$ as defined in Eq. (8). Differentiating \mathcal{L} with respect to θ , the gradient can be found to be:

$$\frac{\partial \mathcal{L}}{\partial \theta} = -\frac{1}{2} \text{tr}(K^{-1} \frac{\partial K}{\partial \theta}) + \frac{1}{2} \alpha^T \frac{\partial K}{\partial \theta} \alpha + \frac{1}{2} \text{tr}(K^{-1} \frac{\partial K}{\partial \theta} K^{-1} \Sigma), \quad (16)$$

where α denotes $K^{-1} \mu$. It is possible to use any gradient-based optimizer to find the hyperparameters. In the experiments, we used a scaled conjugate gradient.

5 Active Preference Exploration

So far, we have assumed that observed preferences are provided in a batch, i.e., they are collected in a rather naive way. Preference prediction, however, could be made more efficient if we can actively select *uncertain* preferences. Consider a typical interaction between a search engine and a user. A user executes a query and considers the results presented. The system now requests the user to provide a set of preference relations in the result with the goal to improve the preference prediction being shown to the user.

Typically, however, the user pays attention to only a few entities in the result. Note that providing preference labels can be quite costly as the user has e.g. to read and understand abstracts of research papers. In turn, important entities may never be considered by the user and no preference feedback is provided. This is likely to lead to suboptimal preference predictions. To avoid this presentation effect [19], we guide the user by asking actively for her preferences among entities so as to provide more useful observations. Usually, we are limited to deploying a small number of preferences only, and thus must carefully choose them.

Being probabilistic models, XPGPs are extremely powerful for active exploration. If we observe a set of preferences corresponding to a finite subset

$\mathcal{A} \subset \mathcal{R}$ of all possibly observable preferences, we can easily predict the uncertainty about any other preference $r \in \mathcal{R} \setminus \mathcal{A}$ conditioned on these observed ones using $P(r|\mathcal{A})$. An approximation of this marginal distribution can be found using the EP method discussed above. It is a Gaussian whose conditional mean $\mu_{r|\mathcal{A}}$ and variance $\sigma_{e_i \succ e_j|\mathcal{A}}^2$ are given by Eqs. (13) and (14) replacing i' with i and j' with j . The values $\hat{\xi}_i$ and $\hat{\xi}_j$ are the overall preference degrees as defined in Eq. (4) of the entities e_i and e_j . Their variances and covariance can be directly obtained from the covariance matrix in Eq. (8) of the approximate posterior distribution $q(\hat{\xi})$.

To solve the active exploration problem, we follow the commonly used greedy approach [15]. That is, we start from an empty set of preferences, $\mathcal{A} = \emptyset$, and greedily add preferences until $|\mathcal{A}| = k$. At each iteration, the greedy rule used is to add the preference $r \in \mathcal{R} \setminus \mathcal{A}$ that has the highest ratio of variance and squared difference in mean, i.e., *the preference with closest latent preference values that we are most uncertain about given the preferences observed so far*. Other scores such as mutual information could also be used [15] but they are out of scope of the present paper.

6 Experimental Analysis

We evaluated the XPGP model for relevance feedback on two real-world datasets, namely OHSUMED and TREC in the LETOR repository [16]. Relevance feedback is an important task in information retrieval: the original answer to user queries are refined based on user feedback (e.g. click or not, browsing time, etc). Thus, the “personalized” ranking list is presented to the user. We used XPGP models to predict preferences on articles (resp. webpages) based on some known preferences. This corresponds to a transductive preference learning. We compared the XPGP model with standard GP [5] and SVM models [13], denoted as PGP and SVM in the experiments. For the GP-based approaches, we used Gaussian kernels, Eq.(1), to compute the covariances on entity attributes and two different graph kernels to obtain correlations on relations: (1) the regularized Laplacian Eq.(2); and (2) Silva *et al.*’s kernel from [21]. For the SVM approach, the radial basis function (RBF) was chosen to compute the kernels. We report on the prediction error rate (ERR) and the area under ROC curve (AUC).

OHSUMED Dataset: The dataset was originally collected by Hersh *et al* [11], and was processed by Liu *et al.* [16] as a benchmark data for learning to rank in information retrieval. As a subset of MEDLINE medical database, the document collection contains 348,566 publications from 270 journals during 1937-1991. Each document consists of title, abstract, MeSH indexing terms, author, source, and publication type. In the dataset, there are 106 queries, each of which are associated with some relevant documents evaluated by humans. The relevance degree has three levels: definitely relevant, partially relevant and not relevant. In total, there are 16,140 query-document pairs. Due to practicality, Liu *et al.* sampled some “possible” relevant documents from the large scale document

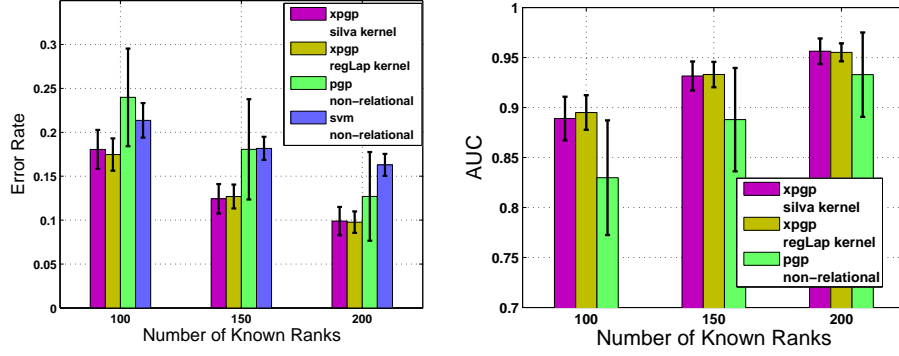


Fig. 2. OHSUMED: Experimental results averaged over 10 queries with 20 random reruns each on predicting preference pairs given different numbers of given preference pairs. **(Left)** Prediction error rate; the lower, the better. **(Right)** AUC values; the larger, the better.

collection and got on average about 152 documents per query. They extracted 25-dimensional feature vectors for each query-document pair. We refer to [16] for more details. The relations between documents are based on similarities, i.e. there is a weighted complete graph between documents; the weight of each edge is the cosine similarity between the contents (keywords) of two documents.

Given a query, each document was originally associated with a relevance degree. Based on this degree we obtained preference pairs of the form $e_i > e_j$. This is not only due to XPGPs modeling assumption but more importantly this information is also more realistic for real-world applications [13]. In the experiments, we randomly selected 100 (150, 200) preference pairs for each query as evidence. Then, the task was to predict the remaining ones. For each setting (100, 150, 200), the selection was repeated 20 times. Note that generating preference pairs was not costly, e.g. one evaluation on relevance provides $n - 1$ preference pairs if there are n documents in a collection.

Fig. 2 shows the experimental results averaged over 10 randomly selected queries. Note that the output of [13] is not “soft” preference pairs, so we report the prediction error rates only. As one can see, in all cases (different number of known preferences), the XPGP model outperforms non-relational GP and SVM models. The significance is demonstrated with Wilcoxon rank sum test (p-value 0.01). The XPGP model performs well, especially when the number of known preference pairs is small. Overall, XPGPs reduced the mean error rates between 12% and 40%. To summarize, modeling relations among entities allows for information to be shared between entities and, in turn, to improve prediction quality. Which graph kernel to use seems to be less important: the two different graph kernels we used performed similarly well. To further verify both results, we compared XPGPs and PGPs on another dataset.

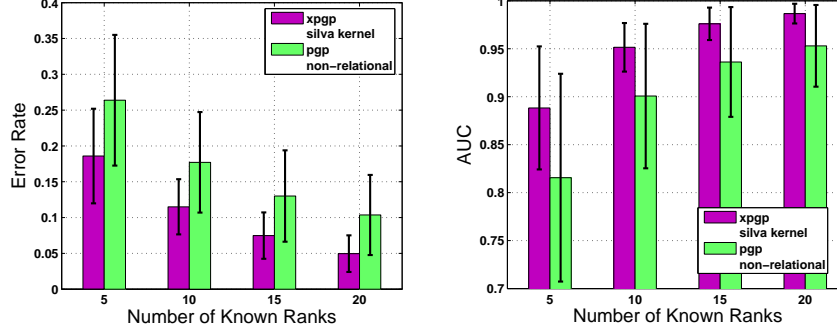


Fig. 3. TREC: Experimental results averaged over 10 queries with 20 reruns each on predicting preference pairs with different numbers of known preference pairs. **(Left)** Prediction error rate; the lower, the better. **(Right)** AUC values; the larger, the better.

TREC Dataset: The TREC dataset was originally collected for a special track on web information retrieval at TREC 2004. The goal of the track was to explore the performance of retrieval methods on large-scale data with hyperlinked structure such as the World Wide Web. The data was crawled from .gov domain in January, 2002. In total, there are 1,053,110 html documents with 11,164,829 hyperlinks. To each query, documents were assigned labels by human experts. Each document has two possible states: relevant/irrelevant. The unlabeled documents are viewed as irrelevant ones. Liu *et al.* [16] processed the TREC dataset and turned it into a benchmark for information retrieval. There were totally 75 queries left. For each query, they ranked the documents with the BM25 scores, and only kept (1) the first 1000 documents and (2) the documents labeled to be relevant. Since the original labels were entity-wise, we converted them into pair-wise ones as follows: we set $e_i \succ e_j$ if e_i is relevant but e_j is irrelevant, $e_i \prec e_j$ otherwise. Again, the processing was not only because of the XPGP setting but because it is more realistic.

In the experiments, we randomly selected 10 queries. On average, there were about 1000 documents associated with each query, which are linked with about 2387 hyperlinks. About 16 of the documents were labeled as relevant. We notice that the hyperlinks are *directed*, i.e. the two webpages involved in a hyperlink play different roles: one is source, the other is target. We convert the directed relations to undirected ones by introducing a relation between two documents, which are linked by the same webpages [21]. For each query, 5 (10, 15, 20) preference pairs were chosen randomly as the known ones. The task was to predict the remaining ones. For each query, the random selection was repeated 20 times, and the mean and the standard deviation of AUC and error rate were computed to measure the prediction performance. Fig. 3 summarizes the experimental results averaged over the 10 queries. As one can see, in all settings (different number of known preferences), the XPGP provides better predictions than the non-relational GP method. A p-value threshold of 0.01 in Wilcoxon

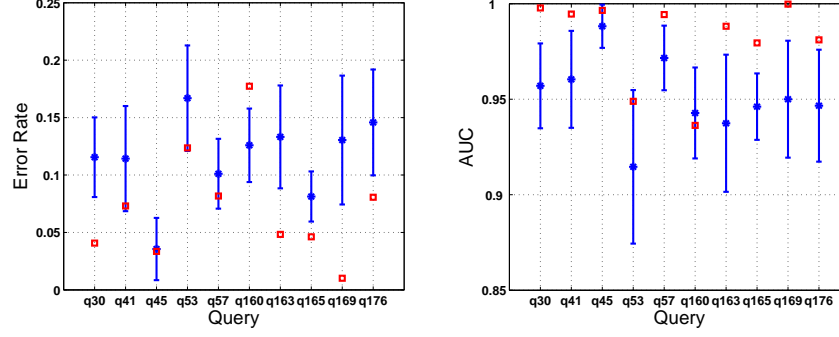


Fig. 4. Actively selecting 10 known preference pairs for 10 TREC queries. The prediction error rate (**left**) and AUC (**right**) values for the naive (means and std. taken from the previous experiment) and for the active selection (red squares) are shown. In 9 out of 10 cases, the active scheme is better than the (mean of the) naive one.

rank sum test showed that the difference is indeed significant. Overall, XPGPs reduced the mean error rates between 30% and 50%.

Active Preference Exploration: So far, the experiments were performed with randomly selected known preferences. In the final experiment, we evaluated the active exploration scheme by automatically selecting 10 known preferences on the TREC dataset. Fig. 4 summarizes the results. As one can see, active selection can indeed improve the prediction quality with higher efficiency than collecting preferences naively.

7 Extensions: Directed, Bipartite and Multiple Relations

Real-world domains typically show multiple relations of different types such as bipartite and directed relations. We will finally show that this situation can easily tackled using XPGPs.

We use distinct latent function values to represent preference factors driven by different types of relations, i.e., we introduce for each entity multiple relational function values, one for each type of relations: $\{g_i^{r_1}, g_i^{r_2}, \dots\}$. The overall preference degree is now the weighted sum of all latent function values associated with the entity: $\xi_i = \omega_1 f_i + \omega_2 g_i^{r_1} + \omega_3 g_i^{r_2} + \dots + \epsilon_i$. We now assume that the latent function values of the same type of relations, e.g. $\{g_1^{r_1}, \dots, g_n^{r_1}\}$, are realizations of random variables in a Gaussian process, i.e., they follow a multivariate Gaussian distribution with mean zero and covariance matrix $K_{r,1}$, which can be specified with graph kernel as discussed before.

In a directed relation, the two involved entities play different roles. Consider e.g. links of webpages **link**: webpage \times webpage. The entities typically serve as the linking and linked webpages. It is reasonable to introduce for each webpage two latent function values representing preference factors for linking and linked

“roles” of the entity. The overall preference degree of a webpage is again a weight sum: $\xi_i = \omega_1 f_i + \omega_2 g_i^{\text{linking}} + \omega_3 g_i^{\text{linked}} + \epsilon_i$. Alternatively, we can use random-walk-based methods to generate kernels on directed graphs [7].

There is only one difficulty when encoding bipartite relations: different types of entities are involved. In turn, graph kernels for univariate relations cannot be used. We address this problem by projecting bipartite relations to univariate ones. Specifically, we add a relation between entities i and j iff. both entities link to the same (heterogeneous) entity. All entities linking to the same (heterogeneous) entity form a clique. Then we can compute the graph kernels on the projected graphs. For example, we can convert a bipartite relation **Direct**: movie \times person to a undirected one **Co-Directed**: movie \times movie.

8 Conclusion

In this paper we have proposed the first nonparametric Bayesian approach to learn preferences from relational data using Gaussian processes. Modeling relations among entities allows for information to be shared between entities and, in turn, to improve prediction quality. The uncertainty model provided by the Gaussian process framework offers predictive uncertainty estimates for preferences, and naturally allowed us to develop an active exploration scheme in which preferences are optimally selected for interactive labeling. Our empirical results showed a significant improvement of preference prediction quality when employing relational information. Furthermore, active preference exploration improved the quality faster than collecting preferences naively.

A natural extension of this work is the adaption of sparse Gaussian processes to the relational case to tackle large-scale datasets. Furthermore, (non)myopic analysis of active preference exploration is interesting as it potentially yields provable bounds on the quality of the estimates. It is likely that other criteria than variance such as mutual information will show better performance.

We believe that this work is an interesting step towards increasing the quality of search engines and information retrieval systems as well as towards well-founded active learning algorithms for relational (preference) learning.

Acknowledgments The authors would like to thank the anonymous reviewers for their comments. They would also like to thank John Guiver, Tie-Yan Liu and Tao Qin for sharing the relational LETOR datasets. Furthermore, they thank Thorsten Joachims for valuable discussions and for encouraging the active exploration scheme. The work was funded by the Fraunhofer ATTRACT Fellowship “Statistical Relational Activity Mining” (STREAM).

References

1. S. Agarwal. Ranking on graph data. In *ICML*, page 2532, 2006.
2. C. Burges, T. Shaked, E. Renshaw, A. Lazier, M. Deeds, N. Hamilton, and G. Hullender. Learning to rank using gradient descent. In *ICML*, 2005.

3. S. Chakrabarti. Learning to rank in vector spaces and social networks. In *WWW*, 2007.
4. W. Chu and Z. Ghahramani. Gaussian processes for ordinal regression. *Journal of Machine Learning Research*, 6:1019–1041, 2005.
5. W. Chu and Z. Ghahramani. Preference learning with gaussian processes. In *ICML*, 2005.
6. W. Chu, V. Sindhwani, Z. Ghahramani, and S. Keerthi. Relational learning with gaussian processes. In *Neural Information Processing Systems*, 2006.
7. F. R. K. Chung. Laplacians and the cheeger inequality for directed graphs. In *Annals of Combinatorics*, volume 9, pages 1–19, 2005.
8. Y. Freund, R. Iyer, R. E. Schapire, and Y. Singer. An efficient boosting algorithm for combining preferences. *JMLR*, 4:933969, 2003.
9. F. Geerts, H. Mannila, and E. Terzi. Relational link-based ranking. In *Proceedings of VLDB-04*, pages 552–563, 2004.
10. J. Guiver and E. Snelson. Learning to rank with softrank and gaussian processes. In *SIGIR*, 2008.
11. W. Hersh, C. Buckley, T. Leone, and D. Hickam. Ohsumed: an interactive retrieval evaluation and new large test collection for research. In *SIGIR*, 2007.
12. E. Hüllermeier, J. Fürnkranz, W. Cheng, and K. Brinker. Label ranking by learning pairwise preferences. *Artificial Intelligence*, 172(16–17):1897–1916, 2008.
13. T. Joachims. Optimizing search engines using clickthrough data. In *SIGKDD*, 2002.
14. H.-C. Kim and Z. Ghahramani. Bayesian gaussian process classification with the em-ep algorithm. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*, volume 28(12), pages 1948–1959, 2006.
15. A. Krause, A. Singh, and C. Guestrin. Near-optimal sensor placements in gaussian processes: Theory, efficient algorithms and empirical studies. *Journal of Machine Learning Research (JMLR)*, 9:235–284, 2008.
16. T.-Y. Liu, T. Qin, J. Xu, W.-Y. Xiong, and H. Li. Letor: benchmark dataset for research on learning to rank for information retrieval. In *SIGIR 2007 Workshop on LR4IR*, 2007.
17. T. Minka. *A family of algorithms for approximate Bayesian inference*. PhD thesis, MIT, 2001.
18. T. Qin, T. Liu, X. Zhang, D. Wang, W. Xiong, and H. Li. Learning to rank relational objects and its application to web search. In *WWW*, 2008.
19. F. Radlinski and T. Joachims. Active exploration for learning rankings from click-through data. In *Proceedings of KDD-07*, pages 570–579, 2007.
20. C. E. Rasmussen and C. K. I. Williams. *Gaussian Processes for Machine Learning*. The MIT Press, 2006.
21. R. Silva, W. Chu, and Z. Ghahramani. Hidden common cause relations in relational learning. In *Neural Information Processing Systems*, 2007.
22. A. J. Smola and I. Kondor. Kernels and regularization on graphs. In *Annual Conference on Computational Learning Theory*, 2003.
23. Z. Xu, K. Kersting, and V. Tresp. Multi-relational learning with gaussian processes. In C. Boutilier, editor, *Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-09)*, 2009. to appear.
24. K. Yu, W. Chu, S. Yu, V. Tresp, and Z. Xu. Stochastic relational models for discriminative link prediction. In *Neural Information Processing Systems*, 2006.
25. X. Zhu, J. Kandola, J. Lafferty, and Z. Ghahramani. Graph kernels by spectral transforms. In O. Chapelle, B. Schoelkopf, and A. Zien, editors, *Semi-Supervised Learning*. MIT Press, 2005.