A GP classification approach to preference learning

Ferenc Huszár
Computational and Biological Learning Lab
Department of Engineering, University of Cambridge
Cambridge, CB2 1PZ

1 Introduction

In this abstract I present the problem of learning from pairwise preference judgements as a special case of binary classification. I discuss why kernel classifiers using traditional kernels based on the distance between items cannot be used to address the problem effectively: the preference prediction problem has inherent symmetry properties that these kernels cannot model. I will review a hierarchical Bayesian model for preference learning that by construction respects these symmetry properties and show its equivalence to probit Gaussian process (GP) classification. Motivated by this model I define the preference judgement kernel and show that it is capable of modelling the symmetry of preference judgement.

The resulting reproducing kernel can be used in conjunction with any standard kernel-based method for classification, notably state-of-the-art large-scale classifiers, or active learning methods. Thus, the main contribution of this work is that a wide range of advanced methods developed for kernel-based classification can now be applied to preference learning practically without changing anything but the kernel. The present work also highlights how hierarchical generative models involving Gaussian process priors can be used to construct meaningful combinations of kernels for non-standard learning problems. Using the same principles it is possible to construct similarly meaningful kernels for more convoluted problems such as multi-user preference learning or multi-task classification.

2 Pairwise preference learning as binary classification

In this paper we consider the problem pairwise preference learning, whereby the dataset consists of pairs of items $(u_k, v_k) \in \mathcal{X}^2$ with associated binary labels $y_k \in \{0, 1\}$; $y_k = 1$ means u_k is preferred to v_k, $y_k = 0$ means v_k is preferred to u_k. I will use the notation $u \succ v$ to denote that u is preferred to v. Our task is to learn, from this set of examples, to predict the order of preference between any pair of items $(u, v) \in \mathcal{X}^2$. There are several approaches to preference learning, but at the end of the day, pairwise preference learning is just an instance of binary classification: for each pair of items (u, v) one the algorithm has to provide a binary label, i. e. our aim is to train a classifier $h : \mathcal{X}^2 \mapsto \{0, 1\}$, or in probabilistic terms to estimate the conditional probability $\mathbb{P}[y = 1|(u, v)]$.

Thus, if we are facing a classification problem, why not try applying a state-of-the art kernel-based classification method, such as a support vector machine (SVM) or Gaussian process classifier (GPC)? Although this is possible, we should be extremely careful for the following reason: traditional kernels, that are based on a distance metric between inputs fail to model a crucial and central symmetry of the preference problem: saying that u is preferred to v is really the same thing as saying that v is not preferred to u. Therefore if (u, v) is labelled as 1 in the dataset, we know that (v, u) should be labelled as 0. Therefore any sensible kernel for this problem would say that (u, v) and (v, u) are maximally dissimilar, no matter what u and v. However, kernels like the popular squared exponential cannot do this, since (u, v) and (v, u) can be arbitrarily far from each other in terms of Euclidean (or other) distance. Therefore, while these kernels of this kind can be used in the context of preference learning, they would be highly ineffective in capturing relevant information from data

and might produce inconsistent preference judgements. The main focus of this paper is then to find meaningful kernels for classification that respect the symmetry properties of preference learning.

To tackle the same problem, the Bayesian learning community has defined hierarchical generative models for preference judgements that by construction respect the degeneracies and symmetries of the problem. In this paper we will focus on the GP-based model presented by [1]. We show that this hierarchical graphical model in equivalent to ordinary GPC with a particularly choice of kernel. This kernel and the corresponding RKHS respects the symmetries of the preference learning problem, thereby providing better generalisation performance than arbitrarily chosen kernels.

The resulting RKHS can be used in conjunction with Gaussian process classifiers as well as other state of the art kernel methods. Most importantly, using the kernel with already existing efficient large-scale classifiers allows us to solve very large scale preference learning problems, that was impossible with state of the art methods.

3 A Gaussian processes-based model of preference learning

I start by briefly reviewing the Gaussian process-based preference learning framework proposed in [1]. The framework assumes the existence of a preference function f over possible items. In a first order approximation we require that, whenever a preference judgement between items u and v is made, u should be preferred if $f(u) > f(v)$. This, however, assumes that the dataset constitutes a proper partially ordered set, which is indeed in conflict with most real-world datasets. Hence the need for a more flexible, probabilistic model, that requires that u is *more likely* to be preferred, if $f(u) > f(v)$. This is achieved in the generative model by assuming that the evaluation of the evaluated preference values $f(u)$ and $f(v)$ is corrupted by random noise every time a judgement is made. The probability of preference can then be written as

$$\mathbb{P}[y = 1|(u_k, v_k), f] = \mathbb{P}[u_k \succ v_k|f] = \mathbb{P}(f(u_k) + \delta_{u_k} > f(v_k) + \delta_{v_k}) = \phi\left(\frac{f(u_k) - f(v_k)}{\sqrt{2}\sigma_\delta}\right) \tag{1}$$

In the above expression, δ_{u_k} and δ_{v_k} denotes the evaluation noise, that are assumed to be independent Gaussian distributed variables with mean 0 and standard deviation σ_δ. For later convenience and without loss of generality, we will assume that $\sqrt{2}\sigma_\delta = 1$. The model is complete with a Gaussian process prior over the latent preference function f:

$$f \sim GP(\mu, k) \tag{2}$$

In the original papers this likelihood model is used in an approximate inference scheme to infer a posterior over the latent preference function f. Note that the likelihood only depends between the difference between $f(u)$ and $f(v)$, $g(u, v) := f(u) - f(v)$. From now on, this function of item-pairs $g : \mathcal{X}^2 \mapsto \mathbb{R}$ will be the main focus of our interest, and we will re-parametrise the inference problem in terms of g. Observe that the likelihood in terms of g is very simple, and in fact equivalent to the *probit classification* likelihood:

$$\mathbb{P}[y = 1|(u_k, v_k), f] = \phi\left(f(u_k) - f(v_k)\right) = \phi\left(g(u_k, v_k)\right) = \mathbb{P}[y = 1|(u_k, v_k), g] \tag{3}$$

Note also that, because g is obtained from f via a linear operation, the Gaussian process prior over f induces a Gaussian process prior over g. The mean μ_{pref}, and covariance function k_{pref} of this GP g can be computed from the mean and covariance of f as follows:

$$\begin{aligned}
k_{pref}((u_i, v_i), (u_j, v_j)) &= Cov[g(u_i, v_i), g(u_j, v_j)] \\
&= Cov\left[\left(f(u_i) - f(v_i)\right), \left(f(u_i) - f(v_i)\right)\right] \\
&= \mathbb{E}\left[\left(f(u_i) - f(v_i)\right) \cdot \left(f(u_i) - f(v_i)\right)\right] - \left(\mu(u_i) - \mu(v_i)\right)\left(\mu(v_j) - \mu(u_i)\right) \\
&= k(u_i, u_j) + k(v_i, v_j) - k(u_i, v_j) - k(v_i, u_j)
\end{aligned}$$

and

$$\begin{aligned}
\mu_{pref}(u, v) &= \mathbb{E}\left[g([u, v])\right] = \mathbb{E}\left[f(u) - f(v)\right] \\
&= \mu(u) - \mu(v)
\end{aligned} \tag{4}$$

We will call k_{pref} the *preference judgement covariance*, or *preference judgement kernel*. We can conclude that this model of predicting preferences between items in $\mathcal{X}$ is equivalent to binary GP classification of items in $\mathcal{X}^2$ with the preference judgement covariance function as follows:

$$g \sim GP(\mu_{pref}, k_{pref}) \tag{5}$$
$$p(y = 1|(u_k, v_k), g) = \Phi(g(u_k, v_k)) \tag{6}$$

Figure 1 illustrates the difference between the original approach where the quantity of central interest was f and our approach where the quantity of interest is g.

4 The preference judgement kernel

It can be shown that the preference judgement kernel, k_{pref} is positive semi-definite. We can also see how k_{pref} respects the symmetry properties of preference learning, by computing the prior correlation between $g(u, v)$ and $g(v, u)$ as follows:

$$\begin{aligned} Corr(g(u,v), g(v,u)) &= \frac{k_{pref}((u,v),(v,u))}{\sqrt{k_{pref}((u,v),(u,v))}\sqrt{k_{pref}((v,u),(v,u))}} \\ &= \frac{k(u,v) + k(v,u) - k(u,u) - k(v,v)}{\sqrt{k(u,u) + k(v,v) - k(u,v) - k(v,u)}\sqrt{k(v,v) + k(u,u) - k(v,u) - k(u,v)}} \\ &= -1 \end{aligned}$$

That is, the value at (u, v) is perfectly anti-correlated with the value at (v, u) under the prior. From this fact it can be shown that all element f of the RKHS corresponding to k_{pref} has the property that f Figure 2 shows samples drawn from a GP prior with the preference learning covariance function. The anti-correlation properties are clearly visible in all of these examples.

5 Extensions

Casting preference learning as a special case of GP classification allows us to adopt several extensions developed for standard GP classification. These include multi-task learning for learning preferences of multiple users, active learning, and sparse approximations to speed up inference on large-scale problems.

6 Experiments

Experiments involving simulated and real-world datasets with various kernel classifiers will be presented at the workshop and in the final paper. Preliminary experiments on the datasets presented in [1] show that expectation propagation for GP classification with the preference kernel outperforms the original inference procedure suggested in [1]. These and other extensions will be discussed in the final paper.

7 Implementation

Reference implementations of the preference judgement kernel/covariance function will be provided for the GPML and LIBSVM toolboxes.

References

[1] Chu, W. & Ghahramani, Z. Preference learning with gaussian processes. in *ICML '05: Proceedings of the 22nd international conference on Machine learning* , 137–144 (ACM, New York, NY, USA, 2005).

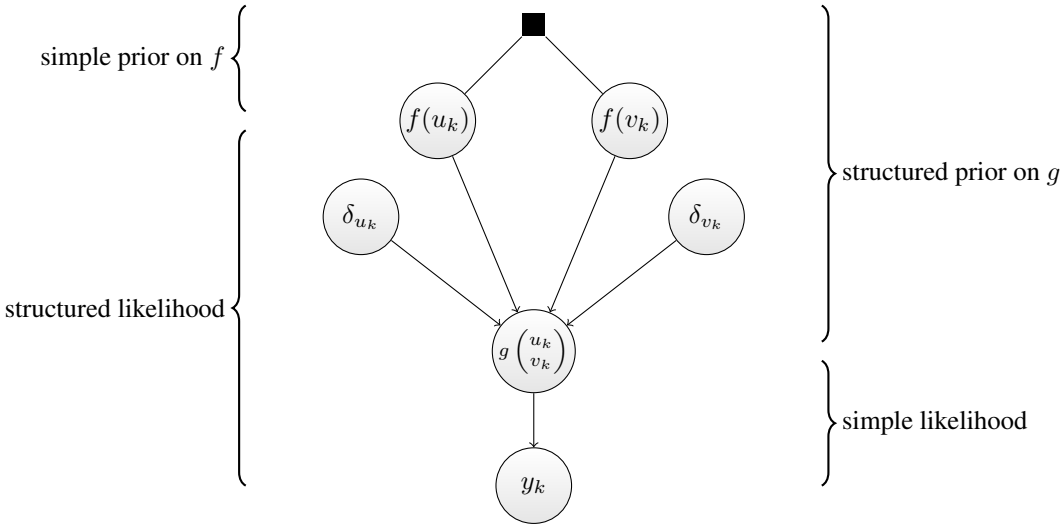

Figure 1: Generative model underlying the preference learning framework. *Left:* the original approach considers the latent preference function f as latent parameter, and the rest of the graphical model as a complex, structured likelihood. *Right:* Our approach re-parametrises the problem in terms of g, and thus works with a simpler likelihood but with a more structured prior. The prior takes the form of a Gaussian process prior with the preference judgement covariance function.

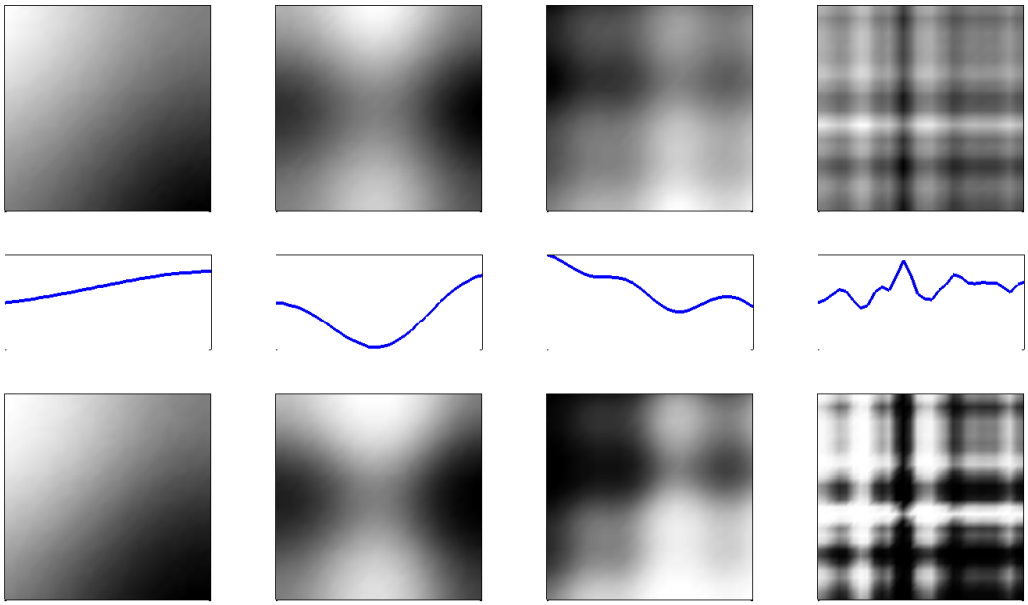

Figure 2: *First row:* Samples drawn from a GP prior with the preference judgement covariance function. The x axis is u, the y axis is v. Brighter colour corresponds to stronger preference for v (note this is the other way round than in the text, I'll change this figure to be in par with the text). The original covariance parameter was a squared exponential with lengthscale decreasing from left to right. *Second row:* f can be reconstructed up to an additive constant from the sampled g as follows: choose an arbitrary v and let $\tilde{f}(u) = g(u, v)$. *Third row:* predictive probabilities $\mathbb{P}[y = 1|(u, v), g]$ corresponding to the sampled functions in the first row (the function is g squashed through the error function Φ). Observe the symmetry properties of the functions: the value at (u, v) is always exactly anti-correlated with the value at (v, u).