Noname manuscript No.
(will be inserted by the editor)

Scalable Bayesian Preference Learning from Crowds

Edwin Simpson · Iryna Gurevych

Received: date

Abstract We show how to make collaborative preference learning work at scale and how it can be used to learn a target preference function from crowdsourced data or other noisy preference labels. The collaborative model captures the reliability of each worker or data source and models their biases and error rates. It uses latent factors to share information between similar workers and a target preference function. We devise an SVI inference schema to enable the model to scale to real-world datasets. Experiments compare results using standard variational inference, laplace approximation and SVI. On real-world data we show the benefit of the personalised model over a GP preference learning approach that treats all labels as coming from the same source, as well as established alternative methods and classifier baselines. We show that the model is able to identify a number of latent features for the workers and for textual arguments.

1 Introduction

Many tasks are more suited to pairwise comparisons than classification etc. Crowds of non-expert annotators may label more accurately if presented with pairs. Implicit feedback may be taken from user actions in an application that can be represented as a preference, such as choosing an option over other options.

There are several works for learning from noisy pairwise comparisons so far (Horvitz et al. 2013 or something like that?). However, these do not provide a way to take account of item features or to model different but valid subjective viewpoints. They assume there is a single ground truth and can therefore

Ubiquitous Knowledge Processing Lab, Dept. of Computer Science, Technische Universität Darmstadt, Germany
E-mail: {simpson,gurevych}@ukp.informatik.tu-darmstadt.de

model only one task and one user's (or a consensus of all users) preferences at once.

Work by Felt et al. 2015, Simpson et al. 2015 etc. shows that item features are particularly useful when combining crowdsourced data. A Gaussian process has not been tested for this purpose before?

GP preference learning presents a way to learn from noisy preferences but assumes constant noise and a single underlying preference function. The collaborative Gaussian process (Houlsby et al. 2012) learns multiple users' preferences. However existing implementations do not scale and do not identify ground truth.

We show how to scale it using SVI and how to use the model to identify ground truth from subjective preferences.

In this paper, we develop methodology to solve the following questions:

1. How can we learn a rating function over large sets of items given a large number of pairwise comparisons?
2. How do we account for the different personal preferences of annotators when inferring the ground truth?

To answer these questions we make the following technical contributions:

1. We propose a method for predicting either gold-standard or personalized ratings by aggregating crowdsourced preference labels using a model of the noise and biases of individual annotators.
2. To enable this method to scale to large, real-world datasets, we develop stochastic variational inference for Bayesian matrix factorization and Gaussian process preference learning.
3. To expedite hyper-parameter tuning, we introduce a technique for gradient-based length-scale optimization of Gaussian processes.

2 Related Work

2.1 Preference Learning from Crowds

¡generic stuff on preference learning including eric horvitz paper, something more recent?¿

These methods assume a single ground truth and model the differences between annotators as noise rather than learning their individual preferences.

Tian et al. Tian and Zhu (2012) consider crowdsourcing tasks where there may be more than one correct answer. They cluster annotators into 'schools of thought' whose members tend to agree with each other. This method assumes that workers are limited to one cluster so cannot model preferences that only partially overlap, such as users who share an interest in a certain genre of books, but whose other interests are different.

2.2 Bayesian Methods for Collaborative Filtering and Preference Learning

A Bayesian approach to preference learning with Gaussian processes, *GPPL*, was introduced by Chu and Ghahramani (2005). This model assumes a single preference function over items, so cannot be used to model the individual preferences of multiple users. The approach was extended by Houlsby et al. 2012 to capture individual preferences using a latent factor model. Pairwise labels from users with common interests help to predict each other's preference function, hence this can be seen as a *collaborative* learning method, as used in *recommender systems*. The inference techniques proposed for this model mean it scales poorly, with computational complexity $\mathcal{O}(N^3 + NP)$, where N is the number of items and P is the number of pairwise labels, and memory complexity $\mathcal{O}(N^2 + NP + P^2)$. In this paper, we address this issue and adapt the model for aggregating crowdsourced data.

2.3 Scalable Bayesian Matrix Factorization

Khan et al. (2014) scalable collaborative GPPL!

2.4 Stochastic Variational Inference

Stochastic variational inference (SVI) is an approximate Bayesian method that addresses the need for scalable inference Hoffman et al. (2013). It has been successfully applied to Gaussian processes Hensman et al. (2015), including Gaussian process classifiers Hensman et al. (2015). It was recently adapted to Gaussian process preference learning Simpson and Gurevych (2018). This paper builds on this work to apply SVI to collaborative Gaussian process preference learning as well as Bayesian matrix factorization in general. We also provide the first full derivation of SVI for GPPL and introduce a technique for efficiently tuning the length-scale of the Gaussian processes.

3 Scalable Bayesian Preference Learning

Following Chu and Ghahramani 2005, we model the relationship between a latent preference function, f, and each observed pairwise label, $v_k \succ u_k$, where k is an index into a list of P pairs, as follows:

$$p(v_k \succ u_k | f(v_k), f(u_k), \delta_{v_k}, \delta_{u_k}) = \begin{cases} 1 & \text{if } f(v_k) + \delta_{v_k} \geq f(u_k) + \delta_{u_k} \\ 0 & \text{otherwise,} \end{cases} \quad (1)$$

where $\delta_i \sim \mathcal{N}(0, 1)$ is Gaussian-distributed noise. The noise term allows for variations in the observed preferences, which may occur if different annotators

disagree or change their minds, or if the preferences are derived from noisy implicit data such as clicks streams. We deviate from Chu and Ghahramani 2005 by assuming δ_i has variance $\sigma = 1$, and instead scale the function f relative to this. This formulation is equivalent but is more convenient for variational inference. We marginalise the noise terms to obtain the preference likelihood:

$$p(v_k \succ u_k | f(v_k), f(u_k)) = \Phi\left(\frac{f(v_k) - f(u_k)}{\sqrt{2}}\right), \tag{2}$$

where Φ is the cumulative distribution function of a standard Gaussian distribution. For the latent function , f, we assume a Gaussian process prior: $f \sim \mathcal{GP}(0, k_\theta/s)$, where k_θ is a kernel function with hyper-parameters θ, and $s \sim \mathcal{G}(a_0, b_0)$ is an inverse scale parameter drawn from a gamma prior with shape a_0 and scale b_0. The kernel function controls the correlation between values of f at different points in the feature space.

The inference goal is to learn the posterior distribution over the function values $f(i)$ for each item i. Chu and Ghahramani 2005 used gradient descent to optimise a Laplace approximation. However, this approach produces a maximum a-posteriori (MAP) approximation, which takes the most probable values of parameters rather than integrating over their distributions in a Bayesian manner and has been shown to perform poorly for tasks such as classification Nickisch and Rasmussen (2008). We address the desire for a better approximation by adapting a variational method based on the extended Kalman filter (EKF) Reece et al. (2011); Steinberg and Bonilla (2014) to the preference likelihood given by Equation 2. To make inference tractable, we approximate the preference likelihood using a Gaussian distribution: $p(v_k \succ u_k | f(v_k), f(u_k)) \approx \mathcal{N}(v_k \succ u_k; \Phi(\hat{f}_{v_k} - \hat{f}_{u_k}/\sqrt{2}), \nu_k)$. The variance v_k is estimated by moment matching with the variance of a beta posterior distribution $\mathcal{B}(p(v_k \succ u_k | f(v_k), f(u_k)); 1 + [v_k \succ u_k], 2 - v_k \succ u_k)$. The values of v_k for all pairs form a diagonal matrix $\boldsymbol{Q}$. This approximation means that the posterior distribution over f for items in the training set is also Gaussian: $p(f(\boldsymbol{x})|\boldsymbol{y}) \approx \mathcal{N}(f(\boldsymbol{x})|\hat{\boldsymbol{f}}, \boldsymbol{C})$ where $\boldsymbol{x}$ is a matrix of input features for the training items. Variational inference is then used to optimise the mean $\hat{\boldsymbol{f}}$ and covariance $\boldsymbol{C}$. by maximising a lower bound, $\mathcal{L}$, on the log marginal likelihood, $p(\boldsymbol{y}|\theta, a_0, b_0)$:

$$\begin{aligned}
\mathcal{L}(q) \approx -\frac{1}{2}\{&L \log 2\pi + \log|\boldsymbol{Q}| - \log|\boldsymbol{C}| + \log|\boldsymbol{K}| \\
&+(\hat{\boldsymbol{f}} - \boldsymbol{\mu})\boldsymbol{K}^{-1}(\hat{\boldsymbol{f}} - \boldsymbol{\mu}) \\
&+(\boldsymbol{y} - \Phi(\hat{\boldsymbol{z}}))^T \boldsymbol{Q}^{-1}(\boldsymbol{y} - \Phi(\hat{\boldsymbol{z}}))\} \\
&+ \Gamma(a) - \Gamma(a_0) + a_0(\log b_0) + (a_0 - a)\widehat{\ln s} \\
&+ (b - b_0)\hat{s} - a \log b,
\end{aligned} \tag{3}$$

where L is the number of observed preference labels, $\boldsymbol{y} = [v_1 \succ u_1, ..., v_L \succ u_L]$ is a vector of binary preference labels, $\widehat{\ln s}$ and $\hat{s}$ are expected values of s, and $\hat{\boldsymbol{z}} = \left\{\frac{\hat{f}_{v_k} - \hat{f}_{u_k}}{\sqrt{2}} \forall k = 1, ..., P\right\}$.

The variational approach described so far requires a scalable inference algorithm. We therefore adapt stochastic variational inference (SVI) Hensman et al. (2013, 2015) to preference learning. For SVI, we assume M *inducing points* with features $\boldsymbol{x}_m$. The inducing points act as a substitute for the complete set of feature vectors of the observed arguments, and allow us to choose $M << N$ to limit the computational and memory requirements. To choose representative inducing points, we use K-means to rapidly cluster the feature vectors, then used the cluster centres as inducing points. Given the inducing points, SVI further limits computational costs by using an iterative algorithm that only considers a subset of the data containing $P_n << P$ pairs at each iteration. The algorithm proceeds as follows:

1. Randomly initialise the mean at the inducing points, $\hat{\boldsymbol{f}}_m$, the covariance of the inducing points, $\boldsymbol{S}$, the inverse function scale expectations $\hat{s}$ and $\hat{\ln s}$, and the Jacobian of the pairwise label probabilities, $\boldsymbol{G}$.
2. Select a random subset of P_n pairwise labels.
3. Compute the mixing coefficient, $\rho_i = (n + \text{delay})^{-\text{forgetting}_\text{rate}}$, which controls the rate of change of the estimates, and the weight $w_n = \frac{P}{P_n}$, which weights each update according to the size of the random subsample.
4. Update each variable in turn using equations below.
5. Repeat from step 2 until convergence.
6. Use converged values to make predictions.

The equations for the updates at iteration n are as follows:

$$\boldsymbol{S}_n^{-1} = (1-\rho_n)\boldsymbol{S}_{n-1}^{-1} + \rho_n \left(\hat{s}\boldsymbol{K}_{mm}^{-1} + w_n \boldsymbol{K}_{mm}^{-1}\boldsymbol{K}_{nm}^T \boldsymbol{G}^T \boldsymbol{Q}^{-1}\boldsymbol{G}\boldsymbol{K}_{nm}\boldsymbol{K}_{mm}^{-T}\right) \tag{4}$$

$$\hat{\boldsymbol{f}}_{m,n} = \boldsymbol{S}_n \left((1-\rho_n)\boldsymbol{S}_{n-1}^{-1}\hat{\boldsymbol{f}}_{m,n-1} + \rho_n w_n \boldsymbol{K}_{mm}^{-1} \boldsymbol{K}_{nm}^T \boldsymbol{G}^T Q^{-1} 1 \left(\frac{1+[v_k \succ u_k]}{3} - \Phi(\hat{\boldsymbol{z}}_n) - \boldsymbol{G}\hat{\boldsymbol{f}}\right)\right) \tag{5}$$

$$\hat{s} = \frac{2a_0 + N}{2b} \tag{6}$$

$$\hat{\ln s} = \Psi(2a_0 + N) - \log(2b) \tag{7}$$

$$\boldsymbol{G} = \frac{1}{2\pi}\exp\left(-\frac{1}{2}\hat{\boldsymbol{z}}_n^2\right) \tag{8}$$

where $\boldsymbol{K}_{mm}$ is the covariance between values at the inducing points, $\boldsymbol{K}_{nm}$ is the covariance between the subsample of pairwise labels and the inducing points, $\hat{\boldsymbol{z}}_n$ is the estimated preference label likelihood for the nth subsample, $b = b_0 + \frac{1}{2}\text{Tr}\left(\boldsymbol{K}_j^{-1}\left(\boldsymbol{\Sigma}_j + (\hat{\boldsymbol{f}}_j - \boldsymbol{\mu}_{j,i})(\hat{\boldsymbol{f}}_j - \boldsymbol{\mu}_{j,i})^T\right)\right)$, and Ψ is the digamma function. Given the converged estimates, we can make predictions for test arguments with feature vectors $\boldsymbol{x}_*$. The posteriors for the latent function values

$\boldsymbol{f}_*$ at the test points have mean and covariance given by:

$$\hat{\boldsymbol{f}}_* = \boldsymbol{K}_{*m}\boldsymbol{K}_{mm}^{-1}\hat{\boldsymbol{f}}_m \tag{9}$$

$$\boldsymbol{C}_* = \frac{\boldsymbol{K}_{**}}{\hat{s}} + \boldsymbol{K}_{*m}\boldsymbol{K}_{mm}^{-1}(\boldsymbol{S} - \boldsymbol{K}_{mm})\boldsymbol{K}_{mm}^{-T}\boldsymbol{K}_{*m}^{T}, \tag{10}$$

where $\boldsymbol{K}_{*m}$ is the covariance between the test items and the inducing points.

3.1 Kernel Length-scale Optimsation

The prior covariance of the latent function f is defined by a kernel k, typically of the form $k_\theta(\boldsymbol{x}, \boldsymbol{x}') = \prod_{d=1}^{D} k_d(|x_d - x'_d|/l_d)$, where k_d is a function of the distance between the values of feature d for item x and x', and a length-scale hyper-parameter, l_d, which controls the smoothness of the function across the feature space. The product over features D may be replaced by other combinations, such as a sum. There are several ways to set l, including the median heuristic Gretton et al. (2012): $l_{d,MH} = \frac{1}{D}\text{median}(\{|x_{i,d} - x_{j,d}|\forall i = 1, .., N, \forall j = 1, ..., N\})$. We can also optimise l_d by choosing the value that maximises the lower bound on the log marginal likelihood, $\mathcal{L}$, defined in Equation 3. This process is known as maximum likelihood II and is often referred to as automatic relevance determination (ARD) Rasmussen and Williams (2006), since features with large length-scales are less relevant because their values have less effect on $k_\theta(\boldsymbol{x}, \boldsymbol{x}')$ than features with short length-scales. The cost of optimisation may be reduced by simultaneously optimising all length-scales using a gradient-based method such as L-BFGS-B Zhu et al. (1997). Given our proposed SVI method, we substitute our inducing point approximation into Equation 3 to approximate the gradients of $\mathcal{L}(q)$ with respect to l_d as follows:

$$\nabla_{l_d}\mathcal{L}(q) = \frac{1}{2}\hat{s}\hat{\boldsymbol{f}}_m^T\boldsymbol{K}_{mm}^{-T}\frac{\partial \boldsymbol{K}_{mm}}{\partial l_d}\boldsymbol{K}_{mm}^{-1}\hat{\boldsymbol{f}}_m$$
$$-\frac{1}{2}\text{tr}\left(\left(\hat{s}\boldsymbol{K}_{mm}^{-1}\boldsymbol{S}\right)^T\left(\boldsymbol{S}^{-1} - \boldsymbol{K}_{mm}^{-1}/\hat{s}\right)\frac{\partial \boldsymbol{K}_{mm}}{\partial l_d}\right) . \tag{11}$$

In our implementation, we choose the Matèrn $\frac{3}{2}$ kernel function for k due to its general properties of smoothness Rasmussen and Williams (2006), so that the matrix of partial derivatives is:

$$\frac{\partial \boldsymbol{K}}{\partial l_d} = \frac{\boldsymbol{K}}{k_d(|\boldsymbol{x}_d, \boldsymbol{x}'_d)}\frac{\partial \boldsymbol{K}_{l_d}}{\partial l_d}, \tag{12}$$

where each entry ij of the $\frac{\partial \boldsymbol{K}_{l_d}}{\partial l_d}$ is defined as:

$$\frac{\partial K_{d,ij}}{\partial l_d} = \frac{3|\boldsymbol{x}_{i,d} - \boldsymbol{x}_{j,d}|^2}{l_d^3}\exp\left(-\frac{\sqrt{3}|\boldsymbol{x}_{i,d} - \boldsymbol{x}_{j,d}|}{l_d}\right). \tag{13}$$

4 Experiments

List of experiments to include:

1. Performance, computation time, memory vs no. inducing points
2. Performance, computation time, memory vs update size
3. Performance, computation time vs different initialisation methods for the inducing points; include different initialisations of K-means

5 Conclusions and Future Work

Acknowledgments

References

Chu W, Ghahramani Z (2005) Preference learning with Gaussian processes. In: Proceedings of the 22nd International Conference on Machine learning, ACM, pp 137–144

Gretton A, Sejdinovic D, Strathmann H, Balakrishnan S, Pontil M, Fukumizu K, Sriperumbudur BK (2012) Optimal kernel choice for large-scale two-sample tests. In: Advances in Neural Information Processing Systems, pp 1205–1213

Hensman J, Fusi N, Lawrence ND (2013) Gaussian processes for big data. In: Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence, AUAI Press, pp 282–290

Hensman J, Matthews AGdG, Ghahramani Z (2015) Scalable Variational Gaussian Process Classification. In: Proceedings of the Eighteenth International Conference on Artificial Intelligence and Statistics, pp 351–360

Hoffman MD, Blei DM, Wang C, Paisley JW (2013) Stochastic variational inference. Journal of Machine Learning Research 14(1):1303–1347

Houlsby N, Huszar F, Ghahramani Z, Hernández-Lobato JM (2012) Collaborative Gaussian processes for preference learning. In: Advances in Neural Information Processing Systems, pp 2096–2104

Khan ME, Ko YJ, Seeger MW (2014) Scalable collaborative bayesian preference learning. In: AISTATS, vol 14, pp 475–483

Nickisch H, Rasmussen CE (2008) Approximations for binary Gaussian process classification. Journal of Machine Learning Research 9(Oct):2035–2078

Rasmussen CE, Williams CKI (2006) Gaussian processes for machine learning. The MIT Press, Cambridge, MA, USA 38:715–719

Reece S, Roberts S, Nicholson D, Lloyd C (2011) Determining intent using hard/soft data and Gaussian process classifiers. In: Proceedings of the 14th International Conference on Information Fusion, IEEE, pp 1–8

Simpson ED, Gurevych I (2018) Finding convincing arguments using scalable bayesian preference learning. Transactions of the Association for Computational Linguistics 6:357–371

Steinberg DM, Bonilla EV (2014) Extended and unscented Gaussian processes. In: Advances in Neural Information Processing Systems, pp 1251–1259

Tian Y, Zhu J (2012) Learning from crowds in the presence of schools of thought. In: Proceedings of the 18th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining, ACM, New York, NY, USA, KDD '12, pp 226–234, DOI 10.1145/2339530.2339571, URL http://doi.acm.org/10.1145/2339530.2339571

Zhu C, Byrd RH, Lu P, Nocedal J (1997) Algorithm 778: L-bfgs-b: Fortran subroutines for large-scale bound-constrained optimization. ACM Transactions on Mathematical Software (TOMS) 23(4):550–560