Personalised Models of Argument Convincingness

Anonymous ACL submission

Abstract

1 Introduction

We hypothesise that different people find different types of argument more convincing than others and therefore, textual features have varying levels of importance in determining convincingness, depending on the audience. We investigate whether certain combinations of textual features are indicative of an argument's convincingness to a particular person. We hypothesise that predictions of convincingness will be more accurate if we adapt the model to the individual reader based on their previously observed preferences. However, preference data for a single individual for any given task can be very sparse, so it will be necessary to consider the similarities between different users' preferences. Furthermore, the computational cost of learning independent models for each person and each task may be impractically high, suggesting a need for more efficient approaches that combine information from multiple users.

Our approach is therefore to identify correlations between different people's preferences so that we can learn shared models of convincingness that can then be adapted to individuals to improve predictions of argument convincingness. We aim to establish whether such a model can be learned by observing pairwise convincingness preferences,

The experiments evaluate a number of techniques for modelling worker preferences, different types of language features, and the correlations between workers and features. We investigate whether workers with similar preferences according to each model give similar justifications for their decisions, thereby lending additional support for models based on correlations between preferences.

We provide a new preference learning model to handle large numbers of potentially very sparse features and large numbers of people. Our Bayesian approach enables us to perform automatic feature selection, learn in semi-supervised or unsupervised modes, and fully account for model and parameter uncertainty, while scaling to large numbers of input features.

2 Related Work

The Gaussian process (GP) preference learning approach of [10] resolves such inconsistencies and provides a way to predict rankings or preferences for items for which we have not observed any pairwise comparisons based on the item's features. An extension to multiple users was proposed by [10], but this method suffered from poor scalability.

Matrix factorisation techniques are commonly used in recommender systems to discover latent user and item features but can fail if the data is very sparse unless suitably regularised or given a Bayesian treatment. Matrix factorisation techniques are also unsuitable for pairwise comparisons as they must be learned using explicit numerical ratings. A more scalable approach that incorporates probabilistic matrix factorisation (specifically, probabilistic PCA) was proposed by [10]. Their method is applicable to both pairwise comparisons and ratings data and as such could be used to learn the model from implicit feedback such as clicks on an item. However, it may be more suitable to use a model for such feedback that explicitly considers the different bias and noise of each type or source of feedback. For such a purpose, the model of [10] may be appropriate but has to date been used for classifier combination and categorical labelling tasks in crowdsourcing and has not been applied to preference learning from different types of feedback. Bayesian approaches are suited to handling these problems of data sparsity, noise and bias, particularly as the modular nature of inference algorithms such as Gibb's sampling and variational approximation is suited to extending the model to handle different types of feedback that give indications of some underlying preferences.

The GP methods require $\mathcal{O}(P_n)$ steps, where P_n is the number of pairs for user n. The method proposed by [10] reduces this scaling issue by using a random sample of pairs at each iteration of their EM algorithm. We use SVI to address scalability in a variational Bayesian framework. The modular nature of VB allows us to take advantage of models for feedback of different types where the input values for each type of feedback do not directly correspond (e.g. explicit user ratings and number of clicks may have different values). By using SVI, we provide a formal way to deal with scalability that comes with guarantees [10]. We also estimate the output scale of the GPs, the latent factors, and item bias as part of the variational approximation.

We compare our work on Sushi-A dataset or against the method of [10] to see if our modifications are actually useful.

Factor analysis differs from PPCA in allowing only diagonal noise covariance matrices, making the observed variables conditionally independent given the latent variables. It also provides a probabilistic treatment for inferring the latent features.

We also investigate whether argumentation preferences can be reduced to a simpler clustering structure, which may be easier to learn with very sparse user data.

3 Identifying Common Patterns of Convincingness

3.1 Baseline methods

- Random: select a label at random
- Most common (MC): select the most common preference label from across the dataset
- No differentiation (ND): we do not model differences between workers. Labels are estimated by taking the average of other people's labels for the same preference pair. When there are no previous pairs available, select the most common preference label
- Gaussian process preference learning with no differentiation (GP-ND): learn a latent ranking function for the objects from pairwise preferences, ignoring differences between workers and features of the arguments. This provides a probabilistic variant of ND

3.2 Modelling Correlations Between Individuals

Two main types of approach:

- Factor analysis – map the set of pairwise preferences to a low-dimensional embedding
- Clustering – assumes that people fall into distinct preference clusters, or can be modelled as a mixture of several archetypes

Specific methods to test can be split into several types. First, we can run different clustering methods on the training data, then predict a worker's label by taking the mean of the other cluster members. When the no members of the cluster have labelled the pair, we predict using the most common label. This method is applied to several clustering algorithms:

- Affinity propagation (AP-mean)
- Gaussian mixture model, using most probable cluster assignment (GMM-mean)
- Gaussian mixture model, using cluster assignments weighted by probability (GMM-WM)

A similar approach can be taken with dimensionality reduction techniques, where we can use K-nearest neightbours (in this case, few workers label each pair, so we choose k=1 and use MC when no workers have labelled the current instance?):

- Factor analysis with K-nearest neighbours (FA-KNN)

Alternatively, we can take a weighted average of the other labels for a pair, where the weights are based on inverse distance from the worker in question in the embedded space:

- Factor analysis with an inverse distance-weighted mean (FA-weighted)

The distance function can be optimised, which leads to proposing more sophisticated methods...

4 Bayesian Preference Learning Model

The model introduced in [10] combines preference learning with matrix factorisation to identify latent features of items and users that affect their preferences. This allows for a collaborative filtering effect, whereby users with similar preferences on a set of observed items are assumed to have similar preferences for other items with similar features. This allows us to make better predictions about the unobserved preferences of a given user when we have seen preferences of a similar user.

The method presented in [10] uses a combination of expectation propagation (EP) and variational Bayes (VB). Since the inference steps require inverting a covariance matrix, this method scales with $\mathcal{O}(N^3)$ and is therefore impractical for large datasets. For our modified version of this method, we improve scalability by using stochastic variational inference to infer the complete model. The variational approximation to the posterior is given by...

The variational inference algorithm maximises a lower bound on the log marginal likelihood:

$$\begin{aligned}\mathcal{L} = &\sum_{i=1}^{N} \mathbb{E}[\log p(t_i | x_{i,1}, x_{i,2}, \boldsymbol{f})] + \\ &\sum_{u=1}^{U} \mathbb{E}\left[\log \frac{p(\boldsymbol{f}_u | \boldsymbol{w}\boldsymbol{y}_u, \boldsymbol{K}_{f,u}/s_{f,u})}{q(\boldsymbol{f}_u)}\right] + \\ &\sum_{c=1}^{C} \mathbb{E}\left[\log \frac{p(\boldsymbol{w}_c | \boldsymbol{0}, \boldsymbol{K}_w/s_{w,c})}{q(\boldsymbol{w}_c)}\right] + \\ &\sum_{c=1}^{C} \mathbb{E}\left[\log \frac{p(\boldsymbol{y}_c | \boldsymbol{0}, \boldsymbol{K}_y/s_{y,c})}{q(\boldsymbol{y}_c)}\right] + \\ &\mathbb{E}\left[\log \frac{p(\boldsymbol{t} | \boldsymbol{\mu}, \boldsymbol{K}_t/s_t)}{q(\boldsymbol{t})}\right] + \\ &\sum_{u=1}^{U} \mathbb{E}\left[\log \frac{p(s_{f,u} | a_{f,u}, b_{f,u})}{q(s_{f,u})}\right] + \\ &\sum_{d=1}^{D} \mathbb{E}\left[\log \frac{p(s_{w,d} | a_{w,d}, b_{w,d})}{q(s_{w,d})}\right] + \\ &\sum_{d=1}^{D} \mathbb{E}\left[\log \frac{p(s_{y,d} | a_{y,d}, b_{y,d})}{q(s_{y,d})}\right] \end{aligned} \tag{1}$$

where t_i is the preference label for the ith pair,

To perform feature selection with large numbers of features, we introduce an automatic relevance determination (ARD) approach that uses the gradient of the lower bound on the log marginal likelihood to optimise the kernel length-scales using the L-BFGS-B method [?]. The gradient is given by:

$$\nabla \mathcal{L} = \left[\frac{\partial \mathcal{L}}{\partial l_{w,1}}, ..., \frac{\partial \mathcal{L}}{\partial l_{w,D_w}}, \frac{\partial \mathcal{L}}{\partial l_{y,1}}, ..., \frac{\partial \mathcal{L}}{\partial l_{y,D_y}} \right], \tag{2}$$

$$\begin{aligned}\frac{\partial \mathcal{L}}{\partial l_{w,d}} = &\frac{\partial}{\partial l_{w,d}} \sum_{u=1}^{U} \mathbb{E}\left[\log \frac{p(\boldsymbol{f}_u | \boldsymbol{w}\boldsymbol{y}_u, \boldsymbol{K}_{f,u}/s_{f,u})}{q(\boldsymbol{f}_u)}\right] + \\ &\sum_{c=1}^{C} \mathbb{E}\left[\log \frac{p(\boldsymbol{w}_c | \boldsymbol{0}, \boldsymbol{K}_w/s_{w,c})}{q(\boldsymbol{w}_c)}\right] - \\ &\sum_{u=1}^{U} \mathbb{E}[\log q(s_{f,u})] - \sum_{d=1}^{D} \mathbb{E}[\log q(s_{w,d})] + \\ = &0.5(\hat{f}_u - wy_u)^T \boldsymbol{K}_{f,u}^{-1} \frac{\partial \boldsymbol{K}}{\partial \log l_{w,d}} \hat{s}_{f,u} \boldsymbol{K}_{f,u}^{-1} (\hat{f}_u - wy_u) \\ &-0.5\mathrm{tr}\left((\boldsymbol{K}_{f,u}^{-1} - \frac{\boldsymbol{C}^{-1}}{\hat{s}_{f,u}}) \frac{\partial \boldsymbol{K}_{f,u}}{\partial \log l_{w,d}} \right) \end{aligned}$$

$$\frac{\partial \mathcal{L}}{\partial l_{y,d}} = \tag{3}$$

$$\tag{4}$$

where $l_{w,d}$ is a length-scale used for all the GPs over item features. The implicit terms are zero when the VB algorithm has converged.

5 Experiments

In the first set of experiments we evaluate the baselines and the different methods for modelling correlations between workers' preferences. In the second set of experiments, we assess the value of different language features. Finally, the third experiment evaluates approaches that integrate both argument features and models of preference correlations.

Prior work on convincingness:

- [10] shows how to predict convincingness of arguments by training a NN from crowd-sourced annotations.
- [10] shows that persuasion is correlated with personality traits.

We build on this to show...

- How we can predict convincingness for a specific user given only previous preferences and preferences of others (collaborative filtering)
- How a combination of text and personality features improves predictions of convincingness

- That we can extract human-interpretable latent features in people and items, which improve performance over just using the input features.

This is all useful because we can use the approach to determine which features are worth obtaining, make predictions when data is sparse, and obtain data from users efficiently.

The steps to show this are:

1. Show a table comparing the baselines, alternative collaborative filtering methods, results from [10], and unsupervised method
2. Add in results when using the input information with out method
3. Show a table comparing the baselines, alternative collaborative filtering methods, results from [10], and unsupervised method
4. Add in results using item information, person information and both
5. Visualise latent features?
6. Table showing importance of input features
7. Add results with lower confidence items excluded to the tables in 1-4. We can also plot the effect of confidence threshold on our results and on the rival methods.
8. Add in Bier/cross entropy – may need to rerun the original code from the previous papers?
9. Run [10] and my complete method with reduced data – check accuracy as it increases. Use confidence cut-off from previous results.
10. Simple active learning approach selecting the most uncertain data point (this will be due to uncertainty about a person, an item with too little data, or disagreement/stochasticity in the likelihood). The plot can be added to the previous results and should be run with rival methods.

6 Future Work

The collaborative preference model can be adapted so that it can be trained using classification data, scores/ratings (a regression task), or a mixture of different observation types by applying a different likelihood. The core of the method is the abstraction of a latent function over items and people, dependent on latent features of items and people, with the ability to include side information and observed features. Future work will therefore investigate the ability to learn from multiple types of labelled data, (rather than only using preference pairs).

A further direction for future work is to apply this model to transfer learning: instead of modelling different latent functions per person, we model latent functions per task. Tasks for which the target function follows a similar pattern would then share information in a collaborative manner, so that training data for one task can inform similar tasks. This may be useful when data is limited, e.g. when performing domain adaptation. In the latter case, there would need to be sufficient similarity between the features of the texts that are being classified for the collaborative effect to take place. For example, in argument mining, we may have several training datasets from different topics, which can be used to learn a model of argument convincingness. Applying a collaborative model would identify topics with common latent features, which would inform predictions on the target domain in parts of the feature space with no training data.

References

W. Chu and Z. Ghahramani, “Preference learning with gaussian processes,” in *Proceedings of the 22nd international conference on Machine learning*. ACM, 2005, pp. 137–144.

N. Houlsby, F. Huszar, Z. Ghahramani, and J. M. Hernández-Lobato, “Collaborative gaussian processes for preference learning,” in *Advances in Neural Information Processing Systems*, 2012, pp. 2096–2104.

M. E. Khan, Y. J. Ko, and M. W. Seeger, “Scalable collaborative bayesian preference learning.” in *AISTATS*, vol. 14, 2014, pp. 475–483.

A. P. Dawid and A. M. Skene, “Maximum likelihood estimation of observer error-rates using the EM algorithm,” *Journal of the Royal Statistical Society. Series C (Applied Statistics)*, vol. 28, no. 1, pp. 20–28, Jan. 1979. [Online]. Available: http://www.jstor.org/stable/2346806

M. D. Hoffman, D. M. Blei, C. Wang, and J. W. Paisley, “Stochastic variational inference.” *Journal of Machine Learning Research*, vol. 14, no. 1, pp. 1303–1347, 2013.

I. Habernal and I. Gurevych, "Which argument is more convincing? Analyzing and predicting convincingness of Web arguments using bidirectional LSTM," in *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Berlin, Germany: Association for Computational Linguistics, 2016, pp. 1589–1599. [Online]. Available: http://www.aclweb.org/anthology/P16-1150

S. Lukin, P. Anand, M. Walker, and S. Whittaker, "Argument strength is in the eye of the beholder: Audience effects in persuasion," in *15th European Chapter of the Association for Computational Linguistics (EACL)*. Association for Computational Linguistics, 2017.