Finding Convincing Arguments using Scalable Bayesian Preference Learning

First Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Second Author
Affiliation / Address line 1
Affiliation / Address line 2
Affiliation / Address line 3
email@domain

Abstract

We introduce a scalable Bayesian preference learning method for identifying convincing arguments in the absence of gold-standard ratings or rankings. In contrast to previous work, we avoid the need for separate approaches or pipelines to produce training data, predict rankings and perform pairwise classification. Bayesian methods are an effective solution when faced with sparse or noisy training data, but have not previously been used to identify convincing arguments. One issue is scalability, which we address by developing a stochastic variational inference method for Gaussian process (GP) preference learning. We show how our method can be applied to predict argument convincingness from crowdsourced data, outperforming state-of-the-art methods, particularly when the data is sparse or noisy. We demonstrate how our Bayesian approach enables more effective active learning, thereby reducing the amount of data required to identify convincing arguments for new users and domains. While word embeddings are principally used with neural networks, our results show that word embeddings in combination with linguistic features also benefit GPs when predicting argument convincingness.

1 Introduction

Arguments are intended to persuade the audience of a particular point of view and are an important way for humans to reason about controversial topics (Mercier and Sperber, 2011). The amount of argumentative text on any chosen topic can, however, overwhelm a reader. Consider the scale of historical text archives and the debates on social media platforms with millions of users. Automated methods could enable readers to overcome this challenge by identifying high-quality, persuasive arguments from different sides of a debate.

Topic: "William Farquhar ought to be honoured as the rightful founder of Singapore".
Stance: "No, it is Raffles!"
Argument 1: HE HAS A BOSS(RAFFLES) HE HAS TO FOLLOW HIM AND NOT GO ABOUT DOING ANYTHING ELSE...
Argument 2: Raffles conceived a town plan to remodel Singapore into a modern city. The plan consisted of separate areas for different...

Figure 1: Example argument pair from an online debate.

Theoretical approaches for assessing argument quality have proved difficult to apply to everyday arguments (Boudry et al., 2015). However, empirical approaches using machine learning have recently shown success in identifying convincing arguments in online discussions (Habernal and Gurevych, 2016). These approaches require examples of arguments paired with judgements of their convincingness. Consider the arguments in Figure 1: how does one assign a numerical convincingness score to each argument? If the audience considers each argument independently, it is difficult to ensure that the scores for all arguments remain consistent with their view of convincingness.

An alternative way to judge arguments is to compare them against one another. In the case of arguments 1 and 2 in Figure 1, we may judge that ar-

gument 1 is less convincing due to its writing style, whereas argument 2 presents evidence in the form of historical events. Pairwise comparisons such as this are known to place less cognitive burden on human annotators than choosing a numerical rating and allow fine-grained sorting of items that is not possible with categorical labels (Kendall, 1948; Kingsley, 2006). Relative judgements also avoid the problem that different annotators may have biases toward high, low or middling ratings, making their scores hard to compare.

In practice, we face a data acquisition bottleneck when encountering new domains or audiences. For example, machine learning methods such as neural networks typically require datasets with many thousands of hand-labelled examples to perform well (Srivastava et al., 2014; Collobert et al., 2011). One solution is to employ multiple non-specialist annotators at low cost (*crowdsourcing*), but this requires quality control techniques to account for errors. Another source of data are the actions of users of a software application, which can be interpreted as pairwise judgements (Joachims, 2002). For example, when a user clicks on an argument in a list it can be interpreted as a preference for the selected argument over more highly-ranked arguments. However, the resulting pairwise labels are an extremely noisy indication of preference.

In this paper, we develop a Bayesian approach to learn from noisy pairwise preferences based on Gaussian process preference learning (GPPL) (Chu and Ghahramani, 2005). We model argument convincingness as a function of textual features, including word embeddings, and develop an inference method for GPPL that scales to realistic dataset sizes using a stochastic variational inference (SVI) (Hoffman et al., 2013). Using datasets provided by Habernal and Gurevych (2016), we show that our method outperforms the previous state-of-the-art for ranking arguments by convincingness and identifying the most convincing argument in a pair. Further experiments show that our Bayesian approach is particularly advantageous with small, noisy datasets, and in an active learning set-up.

The rest of the paper is structured as follows. Section 2 reviews related work on argumentation, then Section 3 motivates the use of Bayesian methods by discussing their successful applications in NLP. In Section 4, we review preference learning methods and then Section 5 describes our scalable Gaussian process-based approach. Section 6 presents our evaluation, comparing our method to the state-of-the art and testing with noisy data and active learning. Finally, we present conclusions and future work.

2 Identifying Convincing Arguments

Lukin et al. (2017) demonstrated that an audience's personality and prior stance affect an argument's persuasiveness, but they were unable to predict belief change accurately. Related work has shown how persuasiveness is also affected by the sequence of arguments in a discussion (Tan et al., 2016; Rosenfeld and Kraus, 2016; Monteserin and Amandi, 2013), but this work focuses on predicting salience given the state of the debate rather than the qualities of arguments.

Habernal and Gurevych (2016) established datasets containing crowdsourced pairwise judgements of convincingness for arguments taken from online discussions. Errors in the crowdsourced data were handled by determining consensus labels using the MACE algorithm (Hovy et al., 2013). The consensus labels were then used to train SVM and bi-directional long short-term memory (BiLSTM) classifiers to predict pairwise labels for new arguments. The MACE consensus labels were also input to PageRank to produce convincingness scores for each argument. These scores were then to train SVM and BiLSTM regression models. A drawback of such pipeline approaches is that they are prone to error propagation (Chen and Ng, 2016), and consensus algorithms such as MACE require multiple crowdsourced labels for each argument pair, which increases annotation costs.

3 Bayesian Methods for NLP

When faced with a lack of annotated data or noisy labels, Bayesian approaches have a number of advantages. Bayesian inference provides a mathematical framework for combining multiple observations with prior information. Given a model, M, and observed data, D, we can apply Bayes' rule to obtain a posterior distribution over M:

$$P(M|D) = \frac{P(D|M)P(M)}{P(D)}. \quad (1)$$

If the dataset is large, the likelihood $P(D|M)$ will dominate the posterior, but when D is small, the posterior will remain closer to the prior, $P(M)$. In contrast, neural network methods typically select model parameters that maximise the likelihood, so are more prone to overfitting with small datasets, which can reduce performance (Xiong et al., 2011).

Bayesian methods can be trained using unsupervised or semi-supervised learning to take advantage of structure in unlabelled data when labelled data is in short supply. Popular examples in NLP are Latent Dirichlet Allocation (LDA) (Blei et al., 2003), which is used for topic modelling, and its extension, the hierarchical Dirichlet process (HDP) (Teh et al., 2005), which learns the the number of topics than requiring it to be fixed a priori. Semi-supervised Bayesian learning has also been used to achieve state-of-the-art results for semantic role labelling (Titov and Klementiev, 2012).

Bayes' rule allows us to combine independent pieces of weak evidence. For instance, a Bayesian network can be used to infer attack relations between arguments by combining votes for acceptable arguments from different people (Hiroyuki Kido, 2017). Graphical models can also combine crowdsourced annotations to train a sentiment classifier without a separate quality control step (Simpson et al., 2015; Felt et al., 2016).

Several successful Bayesian approaches in NLP make use of Gaussian processes (GPs), which are prior distributions over functions of input features. GPs are nonparametric, meaning they can model highly nonlinear functions by allowing function complexity to grow with the amount of data (Rasmussen and Williams, 2006). They account for model uncertainty when extrapolating from sparse training data and can be incorporated into larger graphical models. Example applications include analysing the relationship between a user's impact on Twitter and the text features of their tweets (Lampos et al., 2014), predicting the level of emotion in text (Beck et al., 2014), and estimating the quality of machine translations given source and translated texts (Cohn and Specia, 2013).

4 Preference Learning

Our aim is to develop a Bayesian method for identifying convincing arguments given their features, which can be trained on noisy pairwise labels. Each label, $i \succ j$, states that an argument, i, is more convincing than another argument, j. This type of learning task is a form of *preference learning*, which can be addressed in several ways. A simple approach is to use a generic classifier by obtaining a single feature vector for each pair in the training and test datasets, either by concatenating the feature vectors of the items in the pair or by computing the difference of the two feature vectors, as in SVM-Rank (Joachims, 2002). However, this approach does not produce ranked lists of convincing arguments without predicting a large number of pairwise labels, nor give scores of convincingness.

Alternatively, we can learn an ordering over arguments directly using Mallows models (Mallows, 1957), which define distributions over list permutations. Mallows models can be trained from pairwise preferences (Lu and Boutilier, 2011), but inference is typically costly since the number of possible permutations is $\mathcal{O}(N^2)$, where N is the number of arguments. Modelling only the ordering does not allow us to quantify the difference between arguments at similar ranks.

To avoid the problems of classifier-based and permutation-based methods, we propose to learn a real-valued convincingness function, f, that takes argument features as input and can be used to predict rankings, pairwise labels, or ratings for individual arguments. There are two well established approaches for mapping pairwise labels to real-valued scores: the Bradley-Terry-Plackett-Luce model (Bradley and Terry, 1952; Luce, 1959; Plackett, 1975) and the Thurstone-Mosteller model (Thurstone, 1927; Mosteller, 2006). Based on the latter approach, Chu and Ghahramani (2005) introduced Gaussian process preference learning (GPPL), a Bayesian model that can tolerate errors in pairwise training labels and gains the advantages of a GP for learning nonlinear functions from sparse datasets. However, the inference method proposed by Chu and Ghahramani (2005) has memory and computational costs that scale with $\mathcal{O}(N^3)$, making it unsuitable for real-world text datasets. The next section

explains how we use recent developments in inference methods to develop scalable Bayesian preference learning for argument convincingness.

5 Scalable Bayesian Preference Learning

Following Chu and Ghahramani (2005), we model the relationship between a latent convincingness 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:

$$p(v_k \succ u_k | f(v_k), f(u_k), \delta_{v_k}, \delta_{u_k}) = \begin{cases} 1 & \text{if } f(v_k) + \delta_{v_k} \geq f(u_k) + \delta_{u_k} \\ 0 & \text{otherwise,} \end{cases} \quad (2)$$

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 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 inference goal is to learn the posterior distribution over the function values $f(\mathbf{x})$ for each argument feature vector $\mathbf{x}$. Chu and Ghahramani (2005) used a Laplace approximation, which finds a maximum a-posteriori (MAP) solution that has been shown to perform poorly in many cases (Nickisch and Rasmussen, 2008). Instead, we use a variational approximation to a fully Bayesian approach (Reece et al., 2011; Steinberg and Bonilla, 2014) and adapt this method to the preference likelihood given by Equation 2. Given a set of observed preference pairs, $\boldsymbol{y}$, we assume an approximation, $q(f, s)$, to the true posterior distribution, $p(f, s|\boldsymbol{y}, \theta, a_0, b_0)$. We then update $q(f, s)$ iteratively to maximises a lower bound on the log marginal likelihood, $\mathcal{L} \leq \log p(\boldsymbol{y}|\theta, a_0, b_0)$. This optimisation procedure minimises the Kullback-Leibler divergence of $p(f, s|\boldsymbol{y}, \theta, a_0, b_0)$ from $q(f, s)$, meaning that $q(f, s)$ converges to an approximate posterior.

The variational approach still requires an $\mathcal{O}(N^3)$ matrix inversion, which is impractical with more than a few hundred data points. However, the recent introduction of stochastic variational inference (SVI) (Hoffman et al., 2013; Hensman et al., 2015) means we can adapt our variational inference method to scale to datasets containing at least tens of thousands of arguments and pairwise labels.

SVI assumes M *inducing points*, which act as a substitute for the observed arguments, and considers only a random subset of the data containing P_n pairs at each iteration. By choosing $M << N$ and $P_n << P$, we limit the computational complexity to $\mathcal{O}(M^3 + MP_n)$ and the memory complexity $\mathcal{O}(M^2 + MP_n + P_n^2)$. To choose representative inducing points, we use K-means with $K = M$ to rapidly cluster the feature vectors, then take the cluster centres as inducing points.

A further benefit of GPs is that they enable automatic relevance determination (ARD) to identify informative features, which works as follows. The prior covariance of f is defined by a kernel function 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 items x and x', and a length-scale hyper-parameter, l_d. The length-scale controls the smoothness of the function across the feature space, and can be optimised by choosing the value of l_d that maximises the lower bound on the log marginal likelihood, $\mathcal{L}$. This process is known as maximum likelihood II (Rasmussen and Williams, 2006). Features with larger length-scales after optimisation are less relevant because their values have less effect on $k_\theta(\boldsymbol{x}, \boldsymbol{x}')$. To cut out the cost of optimising the length-scales, we can also use a median heuristic, which has been shown to perform well in practice (Gretton et al., 2012): $l_{d,MH} = \frac{1}{D}\text{median}(\{|x_{i,d} - x_{j,d}| \forall i = 1, .., N, \forall j = 1, ..., N\})$.

6 Experiments

6.1 Datasets

We test our approach on datasets provided by Habernal and Gurevych (2016), which contain pairwise labels for arguments taken from online discussion forums. A pairwise label can have a value of 0, meaning the annotator found the second argument in the pair more convincing, 1 if the annotator was undecided, or 2 if the first argument was more convincing. To test different scenarios, differ-

Dataset	Pairs	Arguments	Undecided	Dataset properties
Toy Datasets	4-13	4-5	0-9	Synthetic pairwise labels Arguments sampled at random from UKPConvArgStrict
UKPConvArg-Strict	11642	1052	0	Combine crowdsourced pairwise labels with MACE Gold labels are $\geq 95\%$ most confident MACE labels Discard arguments marked as equally convincing Discard conflicting preferences
UKPConvArg-Rank	16081	1052	3289	Combine crowdsourced pairwise labels with MACE Gold labels are $\geq 95\%$ most confident MACE labels PageRank run on each topic to produce gold rankings
UKPConvArg-CrowdSample	16927	1052	3698	One original crowdsourced label per pair PageRank run on each topic to produce gold rankings

Table 1: Summary of the internet argument datasets produced using different processing steps.

ent pre-processing steps were used to produce the four datasets shown in Table 1. For all datasets we perform 32-fold cross validation, using 31 folds for training and one for testing. Each fold corresponds to one of 16 controversial topics, and one of two stances for that topic. The toy datasets are used to illustrate the different behaviour of our compared methods (described below). *UKPConvArgStrict* and *UKPConvArgRank* test performance with noise-free labelled data, while *UKPConvArgCrowdSample* is used to evaluate performance with noisy crowdsourced data including conflicts and undecided labels, and to test the suitability of our method for active learning to address the cold-start problem in new domains with no labelled data.

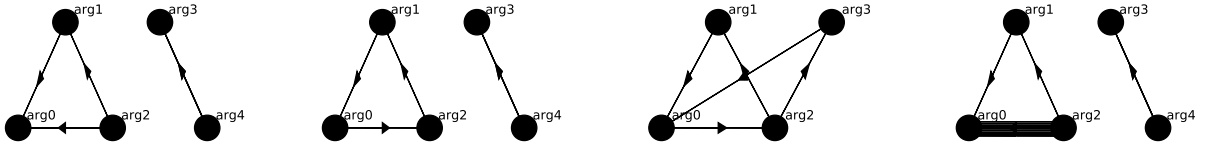
6.2 Method Comparison

Our two basic tasks are *ranking* arguments in terms of convincingness and *classification* of pairwise labels for pairs of arguments, i.e. predicting which argument is preferred. We compare our scalable Gaussian process preference learning method (*GPPL*) against the state-of-the-art SVM approach and a bi-directional long short-term memory network (BiLSTM), both tested by Habernal and Gurevych (2016). For both the classification and ranking tasks, GPPL is trained using the pairwise labels for the training folds. We rank arguments by their expected convincingness, $\mathbb{E}[f(\mathbf{x})]$ for an argument with feature vector $\mathbf{x}$, under the approximate posterior $q(f, s)$. The value of $\mathbb{E}[f(\mathbf{x})]$ is output by our SVI algorithm. Classification probabilities are obtained by substituting $\mathbb{E}[f(\mathbf{x})]$ for $f(v_k)$ or $f(u_k)$ in Equation 2. To apply SVM and BiLSTM to the classification task, we concatenate the feature vectors of each pair of arguments in the training and test sets, and train on the pairwise labels. For ranking, PageRank is first applied to arguments in the training folds to obtain gold-standard scores from the pairwise labels. SVM and BiLSTM regression models are then trained using the PageRank scores.

As a Bayesian alternative to GPPL, we test a Gaussian process classifier (*GPC*) for the classification task by concatenating the feature vectors of arguments in the same way as the SVM classifier. We also evaluate a non-Bayesian approach that uses the same pairwise preference likelihood as GPPL but trains an SVM regression model instead of a GP (*PL+SVR*).

As input features, SVM uses ~ 32000 linguistic features, labelled *ling* in the results, including unigrams, bigrams, ratios and counts of different parts-of-speech and verb forms, dependency tree depth, ratio of exclamation or quotation marks, counts of several named entity types, POS n-grams, presence of dependency tree production rules, readability measures, sentiment scores, spell-checking, and word counts. BiLSTM uses *Glove* word embeddings with 300 dimensions. Both of these feature sets were developed by Habernal and Gurevych (2016).

As word embeddings may contain complementary semantic information to linguistic features, we evaluate GPPL with each feature set and a combination of both, *ling + Glove*. To create a single embedding vector per argument as input for GPPL, we take the mean of individual word embeddings for tokens in the argument. As an alternative, we also tested skip-thoughts (Kiros et al., 2015) and



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided prefs.

Figure 2: Argument preference graphs for each scenario. Arrows point to the preferred argument.

Siamese-CBOW (Kenter et al., 2016) with GPPL on UKPConvArgStrict and UKPConvArgRank, both with ARD optimisation and using the median heuristic, and alone and combined with *ling*. However, we found that mean Glove embeddings produced substantially better performance in all tests.

We set the GPPL hyper-parameters $a_0 = 2$ and $b_0 = 200$ after testing three different settings on UKPConvArgStrict and UKPConvArgRank. This is a weak prior favouring a moderate level of noise in the pairwise labels. For the kernel function, k_d, we chose the Matérn $\frac{3}{2}$ function due to its effectiveness across a wide range of tasks (Rasmussen and Williams, 2006). To set the length-scales, we compare the median heuristic (labelled "medi.") with MLII optimisation (labelled as "OptGPPL").

6.3 Experiment 1: Toy Data

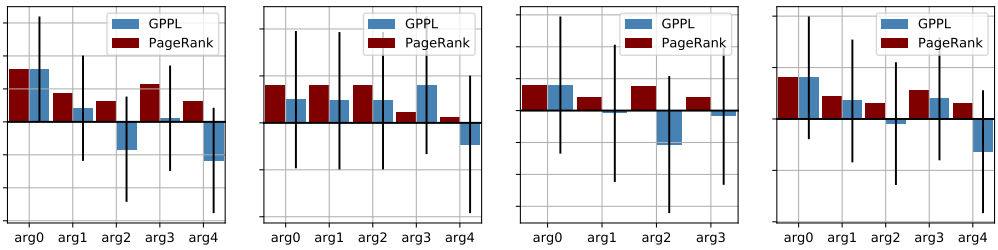
We use synthetic data to illustrate the different behaviour of GPPL, SVM for pairwise classification, and PageRank for scoring arguments. We simulate four scenarios, each of which contains arguments labelled *arg0* to *arg4*. In each scenario, we generate a set of pairwise preference labels according to the convincingness graphs shown in Figure 2. Each scenario is repeated 25 times: in each repeat, we select arguments at random from one fold of UKPConvArgStrict then associate the mean Glove embeddings for these arguments with the labels arg0 to arg4. We train GPPL, PageRank and the SVM classifier on the preference pairs shown in each graph and make predictions for arguments arg0 to arg4.

In the "no cycle" scenario, arg0 is preferred to both arg1 and arg2, which is reflected in the PageRank and GPPL scores in Figure 3. However, arg3 and arg4 are not connected to the rest of the graph and receive different scores with PageRank and GPPL. Figure **??** shows how GPPL provides probabilistic classifications that are less confident for pairs that were not yet observed, e.g. arg2 $\succ$ arg4. This contrasts with Figure 5 which shows discrete classifications produced by SVM.

The "single cycle" scenario shows how each method handles a cycle in the preference graph. Both PageRank and GPPL produce equal values for the arguments in the cycle (arg0, arg1 and arg2). PageRank assigns lower scores to both arg3 and arg4 than the arguments in the cycle, while GPPL more intuitively gives a higher score to arg3, which was preferred to arg4. SVM predicts that arg0 and arg1 are preferred over arg3, although arg0 and arg1 are in a cycle so there is no reason to prefer arg0 and arg1. GPPL, in contrast, gives a weak prediction that arg3 is preferred.

In the "double cycle" scenario, PageRank and GPPL produce very different results. Here, the argument graph shows two paths from arg2 to arg0 via arg1 or arg3, and one conflicting preference arg2 $\succ$ arg0. GPPL scores the arguments as if the single conflicting preference, arg2 $\succ$ arg0, is less important than the two parallel paths from arg2 to arg0. In contrast, PageRank gives high scores to both arg0 and arg2. The classifications by GPPL and SVM are similar, but GPPL produces more uncertain predictions than in the first scenario due to the conflict.

Finally, "cycle with 9 undecided prefs" shows an exaggerated scenario in which we have added nine undecided labels to the "no cycle" scenario, indicated by undirected edges in Figure 2, to simulate a case where multiple annotators labelled the pair and did not all agree. This does not affect the PageRank scores, but reduces the difference in GPPL scores between arg0 and the other arguments, since GPPL gives the edge from arg0 to arg0 less weight due to



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided

Figure 3: Mean scores over 25 repeats. Bars for GPPL show standard deviation of convincingness function posterior.

the undecided labels. This is reflected in the GPPL classifications, which are less confident than in the "no cycle" scenario. The SVM cannot be trained using the uncertain labels and therefore does not adapt to the undecided labels.

In conclusion, GPPL appears to resolve conflicts in the preference graphs in a more intuitive manner than PageRank, which was designed for ranking web pages by importance rather than preference. In contrast to SVM, GPPL is able to account for undecided labels to soften the latent convincingness function.

6.4 Experiment 2: UKPConvArgStrict and UKPConvArgRank

We compare classification performance on UKPConvArgStrict and ranking performance on UKPConvArgRank. Both datasets were cleaned to remove disagreements between annotators as stated in Table 1.

The results are shown in Table 2. When using *ling* features, GPPL produces similar accuracy and improves the area under the ROC curve (AUC) by 2%, and cross entropy error by 0.01. The AUC quantifies how well the predicted probabilities separate the classes, while the cross entropy error quantifies the usefulness of the probabilities output by each method. Much larger improvements can be seen in the ranking metrics. When GPPL is run with mean Glove embeddings, it performs worse than BiLSTM for classification but improves the ranking metrics. Using a combination of features, GPPL outperforms the alternative methods for both classification and ranking, suggesting that embeddings and linguistic features contain complementary information.

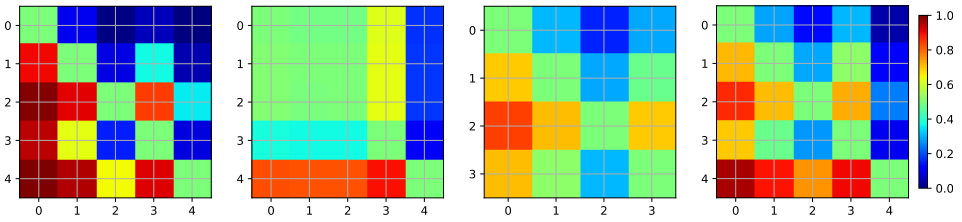
Optimising the length-scale using Bayesian model selection improves classification accuracy by 2% over the median heuristic, giving a statistically significant improvement over SVM, the previous state-of-the-art ($p = 0.0434$ using two-tailed Wilcoxon signed-rank test). The differences in ranking metrics between GPPL opt. *ling* + *Glove* and SVM are highly statistically significant, with $p <<< 0.01$. However, the cost of these improvements is that each fold required around 2 hours to compute instead of approximately 10 minutes on the same machine (an Intel i7 quad core desktop) using the median heuristic.

The results show that PL+SVR does not reach the same performance as GPPL, suggesting that GPPL benefits from integrating a GP in a Bayesian manner. GPC produces the best results on the classification task, indicating the benefits of the Bayesian approach, although it cannot be used to rank the arguments. The classification improvement over GPPL may result from training directly for this task, rather than through a preference learning likelihood. In this experiment, the larger feature space of GPC due to concatenating the feature vectors of the first and second items in each pair does not seem to have damaged its performance.

6.5 Experiment 3: Conflicting and Noisy Data

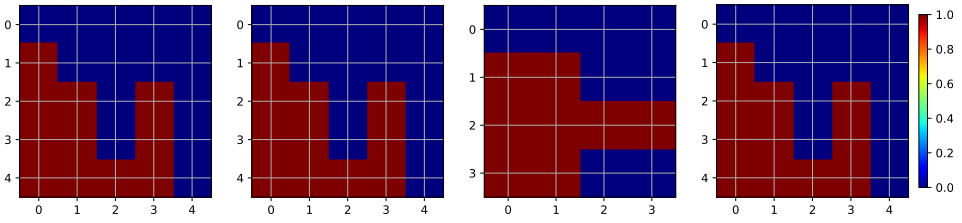
In this experiment, we use UKPConvArgCrowdSample to introduce noisy crowdsourced data including conflicting pairwise labels to both the classification and the regression tasks. Our hypothesis was that GPPL would be better able to handle unreliable crowdsourced data.

The results in Table 3 show that all methods perform worse compared to Experiment 2 due to the noisy pairwise labels. GPPL and GPC produce the best results, but GPC no longer has a clear advan-



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided

Figure 4: Mean GPPL predictions over 25 repeats. Probability that the argument on the horizontal axis is preferred to the argument on the vertical axis.



(a) no cycle (b) single cycle (c) double cycle (d) cycle with 9 undecided

Figure 5: Mean SVM predictions over 25 repeats. Probability that the argument on the horizontal axis is preferred to the argument on the vertical axis.

tage over GPPL. GPPL now outperforms the other methods in all metrics except Spearman's ρ, where PL+SVR performs slightly better. It is possible that GPC and SVM have the largest drops in accuracy compared to the UKPConvArgStrict results because they have no mechanism to resolve conflicts in the preference graph. The performance of the BiLSTM classifier also decreases by a smaller amount, but was already poorer than the other methods on UKPConvArgStrict. PL+SVR is again slightly poorer than GPPL and GPC. Metrics for ranking on UKPConvArgCrowdSample show that while GPPL and PL+SVR continue to perform well, the results for BiLSTM and particularly for SVM are much poorer than on UKPConvArgRank. The differences between GPPL and SVM are highly statistically significant with $p <<< 0.01$ for all classification and ranking metrics, as is the difference in accuracy between GPPL and GPC, and between GPPL and PL+SVR.

UKPConvArgCrowdSample					
	SVM ling	Bi-LSTM Glove	GPPL medi. ling+ Glove	PL+ SVR ling+ Glove	GPC ling+ Glove
Acc:	0.70	0.73	**0.77**	0.75	0.73
AUC:	0.81	0.80	0.84	0.82	**0.86**
CEE:	0.58	0.54	**0.50**	0.55	0.53
Pears.:	0.18	0.26	**0.35**	0.31	-
Spear.:	0.17	0.20	0.54	**0.55**	-
Kend.:	0.12	0.13	**0.40**	**0.40**	-

Table 3: Performance comparison on datasets containing conflicts and noise.

6.6 Experiment 4: Active Learning

In this experiment, we hypothesised that GPPL provides more meaningful confidence estimates than SVM or BiLSTM, which can be used to facilitate active learning in scenarios where labelled training data is expensive or initially unavailable. To test this hypothesis, we simulated an active learning scenario, in which an agent iteratively learns a model for each fold. Initially, $N_{inc} = 2$ pairs were chosen at random from the training set, then used to train the classifier. The agent then performs *uncertainty sampling* to select the $N_{inc} = 2$ pairs with the least confident classifications. The labels for these pairs are then taken from the training set and used to re-train the model. The result is plotted in Figure 6, show-

UKPConvArgStrict								
	SVM	BiLSTM	GPPL medi.			GPPLopt.	GPC	PL+ SVR
	ling	Glove	ling	Glove	ling+ Glove			
Acc.:	0.78	0.76	0.78	0.71	0.79	0.80	**0.81**	0.78
AUC:	0.83	0.84	0.85	0.77	0.87	0.87	**0.89**	0.85
CEE:	0.52	0.64	0.51	1.12	0.47	0.51	**0.43**	0.51
UKPConvArgRank								
Pearson's r:	0.36	0.32	0.38	0.33	**0.45**	0.44	-	0.39
Spearman's ρ :	0.47	0.37	0.62	0.44	0.65	**0.67**	-	0.63
Kendall's τ :	0.34	0.27	0.47	0.31	0.49	**0.50**	-	0.47

Table 2: Performance comparison on UKPConvArgStrict and UKPConvArgRank datasets.

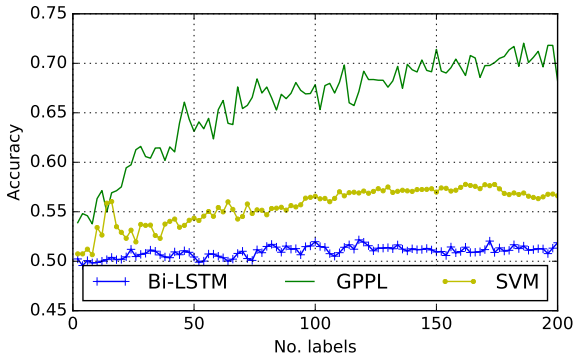


Figure 6: Active learning simulation showing mean accuracy of preference pair classifications over 32 runs.

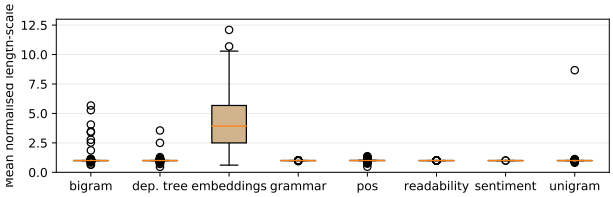


Figure 7: Distribution of length-scales for each type of feature after optimisation. Values are relative to the median heuristic value before optimisation, optimised on fold "should physical education be mandatory in schools – no", where optimisation increased accuracy from 75% to 80%.

ing that GPPL is able to reach accuracies above 65% with only 70 labels, while SVM and BiLSTM do not reach the same performance given 200 labels. The accuracy of GPPL also increases by approximately 17% given 200 labels, while SVM increases approximately 6% and BiLSTM only 2%. This suggest that GPPL may be a more suitable model to be used with uncertainty sampling.

6.7 Relevant Feature Determination

Finally, we show how the length-scales learned by optimising GPPL can be used to identify informative sets of features. A larger length-scale causes greater smoothing, implying that the feature is less relevant when predicting the convincingness function. than a feature with a small length-scale. Figure 7 shows the distribution of normalised length-scales for *ling + Glove* after optimising on one fold of UKPConvArgStrict. Due to the computation time required, our optimisation procedure was limited to 25 function evaluations, which may have resulted in the large number of values close to 1, as features with larger gradients were optimised first.

The length-scales for many dimensions of the mean word embeddings were increased, giving ratios close to 4 times the median heuristic, suggesting that these dimensions may be only very weakly informative. Table 4 shows the largest and smallest ratios for embeddings and linguistic features. The unigram "safety" has a very high length-scale, suggesting it is not informative and may be discarded.

6.8 Error Analysis

We compared the errors when using GPPL opt. with mean Glove embeddings and with linguistic features. We manually inspected the twenty-five arguments most frequently mis-classified by GPPL *ling* and correctly classified by GPPL *Glove*. We found that GPPL *ling* mistakenly marked several arguments as less convincing when they contained grammar and spelling errors but otherwise made a logical

Feature	Ratio
ProductionRule-S->ADVP,NP,VP,.,	0.466
Pos-ngram-PP-O-CARD	0.477
Unigram-“safer”,	0.640
Bigram-“?”-“look”	5.672
Unigram-“safest”	8.673
Unigram-“safety”	271.190
Embedding-dimension-19	0.610
Embedding-dimension-241	12.093

Table 4: Ratios of optimised to median heuristic length-scales: largest and smallest ratios for linguistic features and word embeddings.

point. In contrast, arguments that did not strongly take a side and did not contain language errors were often marked mistakenly as more convincing.

We also examined the twenty-five arguments most frequently misclassified by GPPL *Glove* and correctly labelled by GPPL *ling*. GPPL *Glove* did not correctly mark arguments as less convincing even though they contained multiple exclamation marks and all-caps sentences. Other failures were very short arguments and underrating arguments containing the emotive term 'rape'. The analysis confirms that the different feature sets can identify different aspects of convincingness.

To investigate the differences between our best approach, GPPL opt. *ling* + *Glove*, and the previous best performer, SVM, we manually examined forty randomly chosen false classifications, where one of either *ling* + *Glove* or SVM was correct and the other was incorrect. We found that both SVM and GPPL falsely classified arguments when they were either very short or long and complex, suggesting deeper semantic or structural understanding of the argument may be required. However, SVM also made mistakes where the arguments contained few verbs.

We also compared the rankings produced by GPPL opt. (ling+Glove), and SVM on UKPConvArgRank by examining the 20 largest deviations from the gold standard rank for each method. Arguments underrated by SVM and not GPPL often contained exclamation marks or common spelling errors (likely due to unigram or bigram features). GPPL underrated short arguments with the ngrams “I think”, “why?”, and “don't know”, which were used as part of a rhetorical question rather than to state that the author was uncertain or uninformed. These cases may not be distinguishable using *ling* + *Glove* features.

An expected advantage of GPPL is that it provides more meaningful uncertainty estimates for tasks such as active learning. We examined whether erroneous classifications correspond to more uncertain predictions when using GPPL and SVM when both methods use the *ling* features. For UKPConvArgStrict, the mean Shannon entropy of the pairwise predictions from GPPL was 0.129 for correct predictions and 2.443 for errors, while for SVM, the mean Shannon entropy was 0.188 for correct predictions and 1.583 for incorrect. With both methods, more uncertain predictions correlate with more errors, but the more extreme values for GPPL suggest that its output probabilities more accurately reflect the probability of error than those given by the SVM classifier.

7 Conclusions and Future Work

We presented a novel Bayesian approach to predicting argument convincingness from pairwise labels using Gaussian process preference learning (GPPL). Using recent advances in approximate inference, we developed a scalable algorithm for GPPL model that is suitable for large NLP datasets. Our experiments demonstrated that our method significantly outperforms the state-of-the-art on a benchmark dataset for argument convincingness. We showed that the method performs well with noisy training data, reducing dependence on a quality control pipeline for crowdsourced data. The active learning results show that GPPL is an effective model for cold-start situations with minimal training data. The results also showed that linguistic features and word embeddings provide complementary information, and that GPPL can be used to automatically identify relevant features.

Future work will evaluate our approach on other NLP tasks where reliable classifications may be difficult to obtain, such as learning to classifiy text from implicit user feedback (Joachims, 2002). We also plan to investigate whether the GP preference function can be trained using a combination of classifications and absolute scores as well as pairwise labels.

References

Daniel Beck, Trevor Cohn, and Lucia Specia. 2014. Joint emotion analysis via multi-task gaussian processes. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1798–1803. ACL.

David M Blei, Andrew Y Ng, and Michael I Jordan. 2003. Latent dirichlet allocation. *Journal of machine Learning research*, 3(Jan):993–1022.

Maarten Boudry, Fabio Paglieri, and Massimo Pigliucci. 2015. The fake, the flimsy, and the fallacious: demarcating arguments in real life. *Argumentation*, 29(4):431–456.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Chen Chen and Vincent Ng. 2016. Joint inference over a lightly supervised information extraction pipeline: Towards event coreference resolution for resource-scarce languages. In *AAAI*, pages 2913–2920.

Wei Chu and Zoubin Ghahramani. 2005. Preference learning with gaussian processes. In *Proceedings of the 22nd international conference on Machine learning*, pages 137–144. ACM.

Trevor Cohn and Lucia Specia. 2013. Modelling annotator bias with multi-task gaussian processes: An application to machine translation quality estimation. In *ACL (1)*, pages 32–42.

Ronan Collobert, Jason Weston, Léon Bottou, Michael Karlen, Koray Kavukcuoglu, and Pavel Kuksa. 2011. Natural language processing (almost) from scratch. *Journal of Machine Learning Research*, 12(Aug):2493–2537.

Paul Felt, Eric K Ringger, and Kevin D Seppi. 2016. Semantic annotation aggregation with conditional crowdsourcing models and word embeddings. In *COLING*, pages 1787–1796.

Arthur Gretton, Dino Sejdinovic, Heiko Strathmann, Sivaraman Balakrishnan, Massimiliano Pontil, Kenji Fukumizu, and Bharath K Sriperumbudur. 2012. Optimal kernel choice for large-scale two-sample tests. In *Advances in neural information processing systems*, pages 1205–1213.

John Guiver and Edward Snelson. 2009. Bayesian inference for plackett-luce ranking models. In *proceedings of the 26th annual international conference on machine learning*, pages 377–384. ACM.

Ivan Habernal and Iryna Gurevych. 2016. 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)*, pages 1589–1599, Berlin, Germany. Association for Computational Linguistics.

James Hensman, Nicolò Fusi, and Neil D Lawrence. 2013. Gaussian processes for big data. In *Proceedings of the Twenty-Ninth Conference on Uncertainty in Artificial Intelligence*, pages 282–290. AUAI Press.

James Hensman, Alexander G. de G. Matthews, and Zoubin Ghahramani. 2015. Scalable Variational Gaussian Process Classification. In *AISTATS*.

Keishi Okamoto Hiroyuki Kido. 2017. A bayesian approach to argument-based reasoning for attack estimation. In *Proceedings of the Twenty-Sixth International Joint Conference on Artificial Intelligence, IJCAI-17*, pages 249–255.

Matthew D Hoffman, David M Blei, Chong Wang, and John William Paisley. 2013. Stochastic variational inference. *Journal of Machine Learning Research*, 14(1):1303–1347.

Dirk Hovy, Taylor Berg-Kirkpatrick, Ashish Vaswani, and Eduard H Hovy. 2013. Learning whom to trust with mace. In *HLT-NAACL*, pages 1120–1130.

Thorsten Joachims. 2002. Optimizing search engines using clickthrough data. In *Proceedings of the eighth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 133–142. ACM.

Maurice George Kendall. 1948. *Rank correlation methods.* Griffin.

Tom Kenter, Alexey Borisov, and Maarten de Rijke. 2016. Siamese cbow: Optimizing word embeddings for sentence representations. In *Proceedings of the The 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*.

David C Kingsley. 2006. Preference uncertainty, preference refinement and paired comparison choice experiments. *Dept. of Economics. University of Colorado*.

Ryan Kiros, Yukun Zhu, Ruslan R Salakhutdinov, Richard Zemel, Raquel Urtasun, Antonio Torralba, and Sanja Fidler. 2015. Skip-thought vectors. In *Advances in neural information processing systems*, pages 3294–3302.

Vasileios Lampos, Nikolaos Aletras, Daniel Preoţiuc-Pietro, and Trevor Cohn. 2014. Predicting and characterising user impact on twitter. In *14th Conference of the European Chapter of the Association for Computational Linguistics 2014, EACL 2014*, pages 405–413.

Tyler Lu and Craig Boutilier. 2011. Learning mallows models with pairwise preferences. In *Proceedings of the 28th international conference on machine learning (icml-11)*, pages 145–152.

R Duncan Luce. 1959. On the possible psychophysical laws. *Psychological review*, 66(2):81.

Stephanie Lukin, Pranav Anand, Marilyn Walker, and Steve Whittaker. 2017. 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.

Colin L Mallows. 1957. Non-null ranking models. i. *Biometrika*, 44(1/2):114–130.

Hugo Mercier and Dan Sperber. 2011. Why do humans reason? arguments for an argumentative theory. *Behavioral and brain sciences*, 34(2):57–74.

Ariel Monteserin and Analía Amandi. 2013. A reinforcement learning approach to improve the argument selection effectiveness in argumentation-based negotiation. *Expert Systems with Applications*, 40(6):2182–2188.

Frederick Mosteller. 2006. Remarks on the method of paired comparisons: I. the least squares solution assuming equal standard deviations and equal correlations. In *Selected Papers of Frederick Mosteller*, pages 157–162. Springer.

Hannes Nickisch and Carl Edward Rasmussen. 2008. Approximations for binary gaussian process classification. *Journal of Machine Learning Research*, 9(Oct):2035–2078.

Robin L Plackett. 1975. The analysis of permutations. *Applied Statistics*, pages 193–202.

Tao Qin, Xiubo Geng, and Tie-Yan Liu. 2010. A new probabilistic model for rank aggregation. In *Advances in neural information processing systems*, pages 1948–1956.

C. E Rasmussen and C. K. I. Williams. 2006. Gaussian processes for machine learning. *The MIT Press, Cambridge, MA, USA*, 38:715–719.

Steven Reece, Stephen Roberts, David Nicholson, and Chris Lloyd. 2011. Determining intent using hard/soft data and gaussian process classifiers. In *Information Fusion (FUSION), 2011 Proceedings of the 14th International Conference on*, pages 1–8. IEEE.

Ariel Rosenfeld and Sarit Kraus. 2016. Providing arguments in discussions on the basis of the prediction of human argumentative behavior. *ACM Transactions on Interactive Intelligent Systems (TiiS)*, 6(4):30.

Edwin D Simpson, Matteo Venanzi, Steven Reece, Pushmeet Kohli, John Guiver, Stephen J Roberts, and Nicholas R Jennings. 2015. Language understanding in the wild: Combining crowdsourcing and machine learning. In *Proceedings of the 24th International Conference on World Wide Web*, pages 992–1002. International World Wide Web Conferences Steering Committee.

Nitish Srivastava, Geoffrey E Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: a simple way to prevent neural networks from overfitting. *Journal of machine learning research*, 15(1):1929–1958.

Daniel M Steinberg and Edwin V Bonilla. 2014. Extended and unscented gaussian processes. In *Advances in Neural Information Processing Systems*, pages 1251–1259.

Chenhao Tan, Vlad Niculae, Cristian Danescu-Niculescu-Mizil, and Lillian Lee. 2016. Winning arguments: Interaction dynamics and persuasion strategies in good-faith online discussions. In *Proceedings of the 25th International Conference on World Wide Web*, pages 613–624. International World Wide Web Conferences Steering Committee.

Yee W Teh, Michael I Jordan, Matthew J Beal, and David M Blei. 2005. Sharing clusters among related groups: Hierarchical dirichlet processes. In *Advances in neural information processing systems*, pages 1385–1392.

Louis L Thurstone. 1927. A law of comparative judgment. *Psychological review*, 34(4):273.

Ivan Titov and Alexandre Klementiev. 2012. A bayesian approach to unsupervised semantic role induction. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 12–22. Association for Computational Linguistics.

Maksims Volkovs and Richard S. Zemel. 2014. New learning methods for supervised and unsupervised preference aggregation. *Journal of Machine Learning Research*, 15(1):1135–1176.

Hui Yuan Xiong, Yoseph Barash, and Brendan J Frey. 2011. Bayesian prediction of tissue-regulated splicing using rna sequence and cellular context. *Bioinformatics*, 27(18):2554–2562.