Mitigating Uncertainty in Document Classification

Xuchao Zhang[†‡], **Fanglan Chen**[†], **Chang-Tien Lu**[†], **Naren Ramakrishnan**[†]
[†]Discovery Analytics Center, Virginia Tech, Falls Church, VA, USA
[‡]NEC Laboratories America, Inc, Princeton, NJ, USA
[†]{xuczhang, fanglanc, ctlu, naren}@vt.edu, [‡]xuczhang@nec-labs.com

Abstract

The uncertainty measurement of classifiers' predictions is especially important in applications such as medical diagnoses that need to ensure limited human resources can focus on the most uncertain predictions returned by machine learning models. However, few existing uncertainty models attempt to improve overall prediction accuracy where human resources are involved in the text classification task. In this paper, we propose a novel neural-network-based model that applies a new dropout-entropy method for uncertainty measurement. We also design a metric learning method on feature representations, which can boost the performance of dropout-based uncertainty methods with smaller prediction variance in accurate prediction trials. Extensive experiments on real-world data sets demonstrate that our method can achieve a considerable improvement in overall prediction accuracy compared to existing approaches. In particular, our model improved the accuracy from 0.78 to 0.92 when 30% of the most uncertain predictions were handed over to human experts in "20NewsGroup" data.

1 Introduction

Machine learning algorithms are gradually taking over from the human operators in tasks such as machine translation (Bahdanau et al., 2014), optical character recognition (Mithe et al., 2013), and face recognition (Parkhi et al., 2015). However, some real-world applications require higher accuracy than the results achieved by state-of-the-art algorithms, which makes it difficult to directly apply these algorithms in certain scenarios. For example, a medical diagnosis system (van der Westhuizen and Lasenby, 2017) is expected to have a very high accuracy to support correct decision-making for medical practitioners. Although domain experts can achieve a high performance in these challenging tasks, it is not always feasible to rely on limited and expensive human input for large-scale data sets. Therefore, if we have a model with 70% prediction accuracy, it is intuitive to ask what percentage of the data should be handed to domain experts to achieve an overall accuracy rate above 90%? To maximize the value of limited human resources while achieving desirable results, modeling uncertainty accurately is extremely important to ensure that domain experts can focus on the most uncertain results returned by machine learning models.

Most existing uncertainty models are based on Bayesian models, which are not only time-consuming but also unable to handle large-scale data sets. Deep Neural networks (DNNs) have attracted increasing attention in recent years and have been reported to achieve state-of-the-art performance in various machine learning tasks (Yang et al., 2016; Iyyer et al., 2014). However, unlike probabilistic models, DNNs are still at the early development stage in regards to providing the model uncertainty in their predictions. For those seeking to address the prediction uncertainty in DNNs, it is common to suffer from the following issues on the text classification task. Firstly, few researchers have sought to improve overall prediction performance when only limited human resources are available. Different from existing methods which focus on the value of uncertainty, this problem needs to get domain experts involved in emphasis on the order of the uncertain predictions. For example, the importance of distance between feature representations is neglected by the majority of existing models, but actually this is crucial for improving the order of uncertain predictions, especially during the pre-training of embedding vectors. Moreover, the methods proposed for continuous feature space cannot be applied to discrete text data. For example, adversarial train-

Proceedings of NAACL-HLT 2019, pages 3126–3136
Minneapolis, Minnesota, June 2 - June 7, 2019.

ing is used in some uncertainty models (Goodfellow et al., 2014; Lakshminarayanan et al., 2017; Mandelbaum and Weinshall, 2017). However, due to its dependence on gradient-based methods to generate adversarial examples, the method is not applicable to discrete text data.

In order to simultaneously address all these problems in existing methods, the work presented in this paper adopts a DNN-based approach that incorporates a novel dropout-entropy uncertainty measurement method along with metric learning in the feature representation to handle the uncertainty problem in the document classification task. The study's main contributions can be summarized as follows:

- A novel DNN-based text classification model is proposed to achieve higher model accuracy with limited human input. In this new approach, a reliable uncertainty model learns to identify the accurate predictions with smaller estimated uncertainty.

- Metric learning in feature representation is designed to boost the performance of the dropout-based uncertainty methods in the text classification task. Specifically, the shortened intra-class distance and enlarged inter-class distance can reduce the prediction variance and increase the confidence for the accurate predictions.

- A new dropout-entropy method based on the Bayesian approximation property of Dropout in DNNs is presented. Specifically, we measure the model uncertainty in terms of the information entropy of multiple dropout-based evaluations combined with the de-noising mask operations.

- Extensive experiments on real-world data sets demonstrate that the effectiveness of our proposed approach consistently outperforms existing methods. In particular, the macro-F1 score can be increased from 0.78 to 0.92 by assigning 25% of the labeling work to human experts in a 20-class text classification task.

The rest of this paper is organized as follows. Section 2 reviews related work, and Section 3 provides a detailed description of our proposed model. The experiments on multiple real-world data sets are presented in Section 4. The paper concludes with a summary of the research in Section 5.

2 Related Work

The work related to this paper falls into two sub topics, described as follows.

2.1 Model Uncertainty

Existing uncertainty models are usually based on Bayesian models, which is Traditional Bayesian models such as Gaussian Process (GP), can measure uncertainty of model. However, as a non-parametric model, the time complexity of GP is increased by the size of data, which makes it intractable in many real world applications.

Conformal Prediction (CP) was proposed as a new approach to obtain confidence values (Vovk et al., 1999). Unlike the traditional underlying algorithm, conformal predictors provide each of the predictions with a measure of confidence. Also, a measure of "credibility serves as an indicator of how suitable the training data are used for the classification task (Shafer and Vovk, 2008). Different from Bayesian-based methods, CP approaches obtain probabilistically valid results, which are merely based on the independent and identically distributed assumption. The drawback of CP methods is their computational inefficiency, which renders the application CP not applicable for any model that requires long training time such as Deep Neural Networks.

With the recently heated research on DNNs, the associated uncertainty models have received a great deal of attention. Bayesian Neural Networks are a class of neural networks which are capable of modeling uncertainty (Denker and LeCun, 1990) (Hernández-Lobato and Adams, 2015). These models not only generate predictions but also provide the corresponding variance (uncertainty) of predictions. However, as the number of model parameters increases, these models become computationally more expensive (Wang and Yeung, 2016). Lee et al. proposed a computationally efficient uncertainty method that treats Deep Neural Networks as Gaussian Processes (Lee et al., 2017). Due to its kernel-based design, however, it is not straightforward to apply this to the deep network structures for text classification. Gal and Ghahramani used dropout in DNNs as an approximate Bayesian inference in deep Gaussian processes (Gal and Ghahramani, 2016) to mitigate the problem of representing uncertainty in deep learning without sacrificing the computational complexity. Dropout-based methods have also been extended to various tasks such as computer vision (Kendall and Gal, 2017), autonomous vehicle safety (McAllister et al., 2017) and medical decision making (van der Westhuizen and Lasenby, 2017).

However, few of these methods are specifically designed for text classification and lack of considerations on improving the overall accuracy in the scenario that domain experts can be involved in the process.

2.2 Metric Learning

Metric learning (Xing et al., 2003; Weinberger et al., 2006) algorithms design distance metrics that capture the relationships among data representations. This approach has been widely used in various machine learning applications, including image segmentation (Gong et al., 2013), face recognition (Guillaumin et al., 2009), document retrieval (Xu et al., 2012), and collaborative filtering (Hsieh et al., 2017). Weinberger et al. proposed a large margin nearest neighbor (LMNN) method (Weinberger et al., 2006) in learning a metric to minimize the number of class impostors based on pull and push losses. However, as yet there have been no report of work focusing specifically on mitigating prediction uncertainties. Mandelbaum and Weinshall (Mandelbaum and Weinshall, 2017) measured model uncertainty by the distance when comparing to the feature representations in training data, but this makes the uncertainty measurement inefficient because it requires an iteration over the entire training data set. To the best of our knowledge, we are the first to apply metric learning to mitigate model uncertainty in the text classification task. We also demonstrate that metric learning can be applied to dropout-based approaches to improve their prediction uncertainty.

3 Model

In this section, we propose a DNN-based approach to predict document categories with high confidence for the accurate predictions and high uncertainty for the inaccurate predictions. The overall architecture of the proposed model is presented in Section 3.1. The technical details for the metric loss and model uncertainty predictions are described in Sections 3.2 and 3.3, respectively.

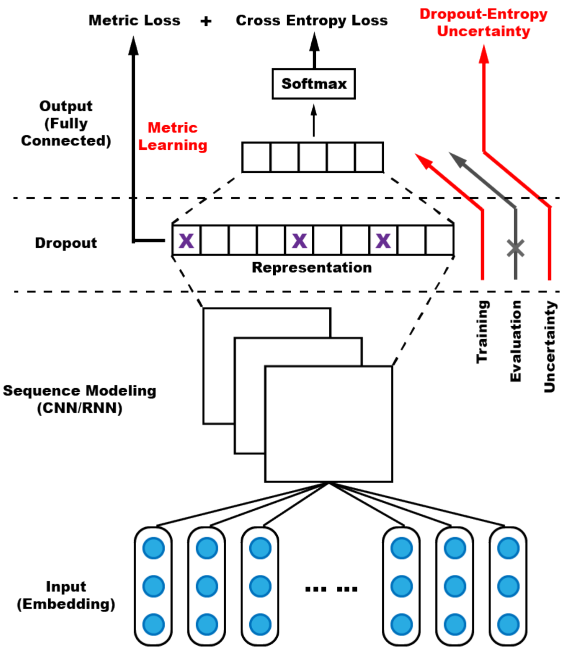


Figure 1: Overall Architecture of Proposed Model

3.1 Model Overview

In order to measure the uncertainty of the predictions for document classification task, we propose a neural-network-based model augmented with dropout-entropy uncertainty measurement and incorporating metric learning in its feature representation. The overall structure of the proposed model is shown in Figure 1. Our proposed model has four layers: 1) **Input Layer**. The input layer is represented by the word embeddings of each words in the document. By default, all word vectors are initialized by Glove (Pennington et al., 2014) pre-trained word vectors in Wikipedia with an embedding dimension of 200. 2) **Sequence Modeling Layer**. The sequence modeling layer extracts the feature representations from word vectors. This is usually implemented by Convolutional Neural Networks (CNN) or Recurrent Neural Networks (RNN). In this paper, we focus on a CNN implementation with max pooling that utilizes 3 kernels with filter sizes of 3, 4 and 5, respectively. After that, a max pooling operation is applied on the output of sequence model. 3) **Dropout layer**. The convolutional layers usually contain a relatively small number of parameters compared to the fully connected layers. It is therefore reasonable to assume that CNN layers suffer less from overfitting, so Dropout is not usually used after CNN layers as it achieves only a trivial performance

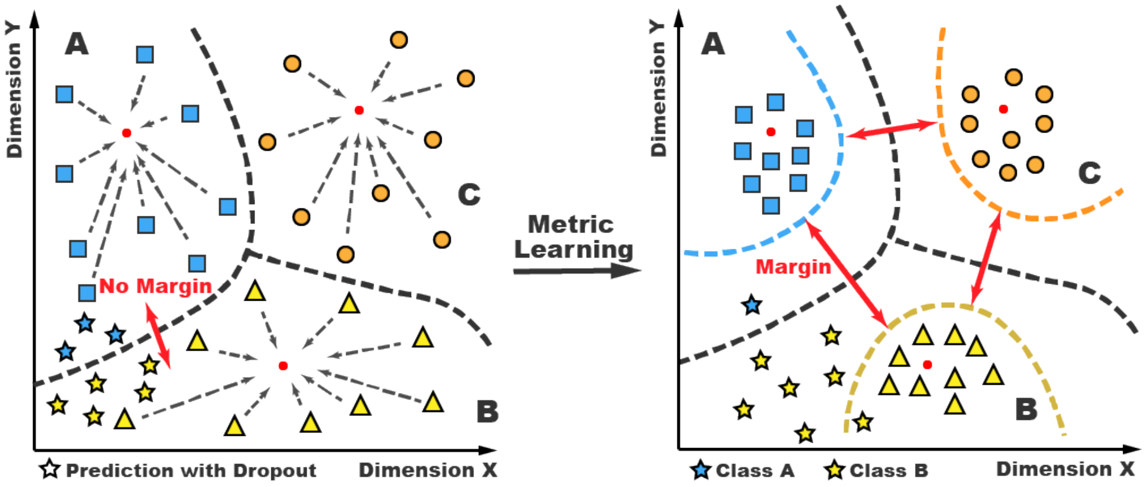


Figure 2: Feature representations with no metric learning (left) and metric learning (right).

improvement (Srivastava et al., 2014). However, since there is only one fully-connected layer in our model, we opted to add one Dropout layer after the CNN layer, not only to prevent overfitting, but also to measure prediction uncertainty (Gal and Ghahramani, 2016). The Dropout operation will be randomly applied to the activations during the training and uncertainty measurement phrases, but will not be applied to the evaluation phrase. 4) **Output layers**. The output is connected by a fully connected layer and the softmax. The loss function of our model is the combination of the cross entropy loss of the prediction and the metric loss of the feature representation. We regard the output of the Dropout layer as the representation of the document and deposit it into a metric loss function. The purpose here is to penalize large distance feature representations in the same class and small distance feature representations among different classes. The details of the metric loss function will be described in Section 3.2.

3.2 Metric Learning on Text Features

For uncertainty learning in text feature space, our purpose is to ensure the Euclidean distance between intra-class instances is much smaller than the inter-class instances. To achieve this, we use metric learning to train the desirable embeddings. Specifically, let $\boldsymbol{r}_i$ and $\boldsymbol{r}_j$ be the feature representations of instances i and j, respectively, then the Euclidean distance between them is defined as $\mathcal{D}(\boldsymbol{r}_i, \boldsymbol{r}_j) = \frac{1}{d}\|\boldsymbol{r}_i - \boldsymbol{r}_j\|_2^2$, where d is the dimension of the feature representation.

Suppose the data instances in the training data contain n classes and these are categorized into n subsets $\{S_k\}_{k=1}^n$, where S_k denotes the set of data instances belong to class k. Then the intra-class loss penalizes the large Euclidean distance between the feature representations in the same class, which can be formalized as Equation (1).

$$\mathcal{L}_{\text{intra}}(k) = \frac{2}{|S_k|^2 - |S_k|} \sum_{i,j \in S_k, i<j} \mathcal{D}(\boldsymbol{r}_i, \boldsymbol{r}_j) \tag{1}$$

where $|S_k|$ represents the number of elements in set S_k. The loss is the sum of all the feature distances between each possible pair in the same class set. Then, the loss is normalized by the number of unique pairs belonging to each class set.

The inter-class loss ensures large feature distances between different classes, which is formally defined as Equation (2).

$$\mathcal{L}_{\text{inter}}(p, q) = \frac{1}{|S_p| \cdot |S_q|} \sum_{i \in S_p, j \in S_q} \Big[m - \mathcal{D}(\boldsymbol{r}_i, \boldsymbol{r}_j)\Big]_+ \tag{2}$$

where m is a metric margin constant to distinguish between the intra- and inter-classes and $[z]_+ = \max(0, z)$ denotes the standard hinge loss. If the feature distance between instances from different classes is larger than m, the loss is zero. Otherwise, we use the value of m minus the distance as its penalty loss, with a larger m representing a larger inter-class distance. This parameter usually varies when we use different word embedding methods. In our experiment, we found that a small m is normally needed when the word embedding is initialized by a pre-trained word vector method such as Glove (Pennington et al., 2014); a larger m is required if word vectors are initialized randomly. The overall metric loss function is defined in Equation (3). This combines the intra-class loss and inter-class loss for all the classes.

$$\mathcal{L}_{\text{metric}} = \sum_{k=1}^{n} \left\{ \mathcal{L}_{\text{intra}}(k) + \lambda \sum_{i \neq k} \mathcal{L}_{\text{inter}}(k, i) \right\} \tag{3}$$

where λ is a pre-defined parameter to weight the importance of the intra- and inter-class losses. We set λ to 0.1 by default.

Figure 2 illustrates an example of a three-class feature representation in two dimensions. The left-hand figure shows the feature distribution trained with no metric learning. Obviously, the feature distance of the intra-class is large, sometimes even exceeding those of the inter-class distance near the decision boundary. However, the features trained by metric learning, shown in the right-hand figure, exhibit clear gaps between the inter-class predictions. This means the predictions with dropout are less likely to result in an inaccurate prediction and even reduce the variance of dropout prediction trials. The example shown in Figure 2 has eight dropout predictions, three of which are classified to an inaccurate class when no metric learning is applied compared to only one inaccurate prediction with metric learning.

3.3 Uncertainty Measurement

Bayesian models such as the Gaussian process (Rasmussen, 2004) provide a powerful tool to identify low-confidence regions of input space. Recently, Dropout (Srivastava et al., 2014), which is used in deep neural networks, has been shown to serve as a Bayesian approximation to represent the model uncertainty in deep learning (Gal and Ghahramani, 2016). Based on this work, we propose a novel information-entropy-based dropout method to measure the model uncertainty in combination with metric learning for text classification. Given an input data instance $\boldsymbol{x}^*$, we assume the corresponding output of our model is y^*. The output computed by our model incorporates a dropout mechanism in its evaluation mode, which means the activations of intermediate layers with Dropout are not reduced by a factor. When we repeat the process k times, we obtain the output vector $\boldsymbol{y}^* = \{y_1^*, \ldots, y_k^*\}$. Note that the outputs are not the same since the output here is generated by applying dropout after the feature representation layer in Figure 1.

Given the output $\boldsymbol{y}^*$ of k trials with Dropout, our proposed uncertainty method has the following four steps, as shown in Figure 3: (1) **Bin count**. We use bin count to calculate the frequency of each class. For example, if the class 2 appears 24 times in the dropout output vector $\boldsymbol{y}^*$, the bin count for class 2 is 24. (2) **Mask**. We use the mask step to avoid random noises in the frequency vector. In this step, we set the largest m elements to have their original values and the remaining ones to zero. The value of m is usually chosen to be $2/3$ of the total class number when the total classes are over 10; otherwise, we just skip the step. (3) **Normalization**. We use the normalization step to calculate the probabilities of each class. (4) **Information entropy**. The information entropy is calculated by $u = -\sum_{i=1}^{c} p_k(i) \log p_k(i)$, where $p_k(i)$ represents the frequency probability of the i-th class in a total k trials and c is the number of classes. We use the entropy value as the uncertainty score here, in which the smaller the entropy value is, the more confident the model is in the output. Take the case in Figure 3 as an example. When the frequency of class 2 is 24, the entropy is 1.204. If the output of the 50 trials all belong to class 2, the entropy becomes 0.401, which means that the model is less uncertain about the predictive results.

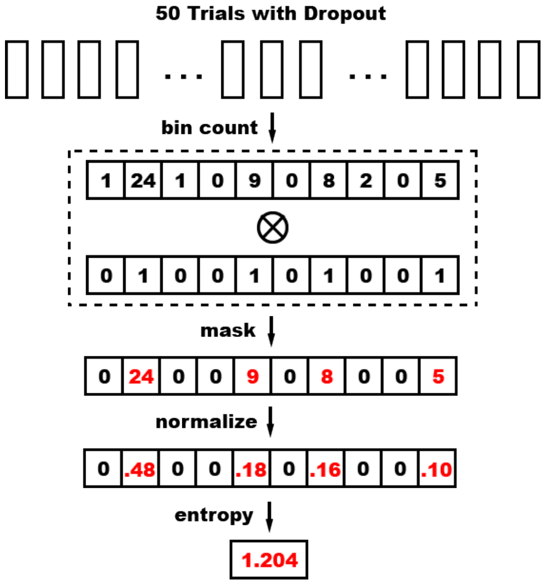


Figure 3: Example of the dropout-entropy method.

4 Experiment

In this section, the performance of the proposed model uncertainty approach is evaluated on multiple real-world document classification data sets.

	Uncertainty Ratio (Micro F1, Improved Ratio)				
	0%	**10%**	**20%**	**30%**	**40%**
PL-Variance	0.760	0.799(5.10%)	0.815(7.30%)	0.827(8.87%)	0.840(10.52%)
Distance	0.780	0.784(0.59%)	0.787(0.94%)	0.795(1.91%)	0.800(2.58%)
NNGP	0.637	0.659(3.44%)	0.670(5.04%)	0.678(6.44%)	0.689(8.11%)
Dropout	0.758	0.792(4.47%)	0.827(9.08%)	0.851(12.18%)	0.879(15.93%)
Dropout + Metric	0.781	0.823(5.38%)	0.863(10.53%)	0.892(14.31%)	0.921(18.05%)
DE	0.760	0.807(6.25%)	0.849(11.73%)	0.888(16.79%)	0.917(20.70%)
DE + Metric	0.781	**0.835(6.93%)**	**0.878(12.47%)**	**0.918(17.62%)**	**0.944(20.92%)**
	Uncertainty Ratio (Macro F1, Improved Ratio)				
	0%	**10%**	**20%**	**30%**	**40%**
PL-Variance	0.751	0.789(5.05%)	0.806(7.24%)	0.818(8.87%)	0.830(10.49%)
Distance	0.773	0.777(0.48%)	0.779(0.81%)	0.786(1.76%)	0.789(2.15%)
NNGP	0.624	0.647(3.56%)	0.657(5.27%)	0.665(6.54%)	0.675(8.13%)
Dropout	0.749	0.781(4.22%)	0.813(8.46%)	0.833(11.10%)	0.860(14.74%)
Dropout + Metric	0.773	0.816(5.47%)	0.853(10.33%)	0.878(13.59%)	0.906(17.14%)
DE	0.752	0.796(5.96%)	0.835(11.05%)	0.872(16.04%)	0.900(19.70%)
DE + Metric	0.774	**0.826(6.70%)**	**0.866(11.97%)**	**0.904(16.87%)**	**0.929(20.02%)**

Table 1: Uncertainty Scores for the 20 NewsGroup Dataset (20 Categories)

After an introduction of the experiment settings in Section 4.1, we compare the performance achieved by the proposed method against those of existing state-of-the-art methods, along with an analysis of the parameter settings and metric learning in Section 4.2. Due to space limitation, the detailed experiment results on different sequence models can be accessed in the full version here[1]. The source code can be downloaded here[2].

4.1 Experimental Setup

In our experiments, all word vectors are initialized by pre-trained Glove (Pennington et al., 2014) word vectors, by default. The word embedding vectors are pre-trained in Wikipedia 2014 with a word vector dimension of 200. We trained all the DNN-based models with a batch size of 32 samples with a momentum of 0.9 and an initial learning rate of 0.001 using the Adam (Kingma and Ba, 2014) optimization algorithm.

4.1.1 Datasets and Labels

We conducted experiments on three publicly available datasets: 1) **20 Newsgroups**[3] (Lang, 1995): The data set is a collection of 20,000 documents, partitioned evenly across 20 different news groups; 2) **IMDb Reviews** (Maas et al., 2011): The data set contains 50,000 popular movie reviews with binary positive or negative labels from the IMDb website; and 3) **Amazon Reviews** (McAuley and Leskovec, 2013): The dataset is a collection of reviews from Amazon spanning the time period from May 1996 to July 2013. We used review data from the Sports and outdoors category, with 272,630 data samples and rating labels from 1 to 5.

For all three data sets, we randomly selected 70% of the data samples as the training set, 10% as the validation set and 20% as the test set.

4.1.2 Evaluation Metrics

In order to answer the question ”*What percentage of data should be transferred to domain experts to achieve an overall accuracy rate above 90%?*”, we measure the classification performance in terms of various uncertainty ratios. Specifically, assuming the entire testing set S has size n and an uncertainty ratio r, we can remove the most uncertain samples S_r from S based on the uncertainty ratio r, where the size of the uncertainty set S_r is $r \cdot n$. We assume the uncertain samples S_r handed to domain experts achieve 100% accuracy. If the uncertainty ratio r equals to 0, the model performs

[1] https://xuczhang.github.io/papers/naacl19_uncertainty_full.pdf

[2] https://github.com/xuczhang/UncertainDC

[3] http://qwone.com/~jason/20Newsgroups/

	Uncertainty Ratio (Accuracy, Improved Ratio)				
	0%	**10%**	**20%**	**30%**	**40%**
PL-Variance	0.878	0.911(3.69%)	0.937(6.70%)	0.955(8.71%)	0.970(**10.42%**)
Distance	0.884	0.893(0.95%)	0.892(0.91%)	0.893(1.04%)	0.895(1.24%)
Dropout	0.880	0.912(3.72%)	0.936(6.43%)	0.957(8.75%)	0.969(10.20%)
Dropout + Metric	0.884	0.917(3.73%)	**0.944**(6.78%)	**0.961**(8.70%)	**0.973**(10.11%)
DE	0.878	0.911(3.70%)	0.937(6.71%)	0.956(8.83%)	0.969(10.33%)
DE + Metric	0.883	**0.918(3.91%)**	**0.944(6.87%)**	**0.961(8.78%)**	**0.973**(10.20%)

	Uncertainty Ratio (F1 Score, Improved Ratio)				
	0%	**10%**	**20%**	**30%**	**40%**
PL-Variance	0.880	0.913(3.68%)	0.939(6.67%)	0.956(8.65%)	0.971(**10.34%**)
Distance	0.885	0.894(1.07%)	0.898(1.42%)	0.901(1.84%)	0.904(2.13%)
Dropout	0.881	0.914(3.70%)	0.938(6.41%)	0.958(8.67%)	0.971(10.13%)
Dropout + Metric	0.885	0.917(3.70%)	**0.944**(6.74%)	**0.961**(8.67%)	**0.974**(10.06%)
DE	0.880	0.913(3.67%)	0.939(6.67%)	0.957(8.77%)	0.970(10.25%)
DE + Metric	0.884	**0.918(3.88%)**	**0.944(6.83%)**	**0.961(8.73%)**	**0.974**(10.14%)

Table 2: Uncertainty Scores for the IMDb Dataset (2 Categories)

	Uncertainty Ratio (Accuracy, Improved Ratio)				
	0%	**10%**	**20%**	**30%**	**40%**
PL-Variance	0.700	0.738(5.43%)	0.764(9.14%)	0.784(1.20%)	0.801(14.4%)
Distance	0.697	0.699(0.29%)	0.702(0.72%)	0.704(1.00%)	0.705(1.15%)
Dropout	0.700	0.735(5.00%)	0.764(9.14%)	0.800(14.29%)	0.831(18.71%)
Dropout + Metric	0.710	0.746(5.07%)	0.779(9.72%)	0.815(14.79%)	0.847(19.30%)
DE	0.700	0.739(**5.57%**)	0.773(10.43%)	0.806(15.14%)	0.836(19.43%)
DE + Metric	0.724	**0.764**(5.52%)	**0.800(10.50%)**	**0.834(15.19%)**	**0.866(19.61%)**

Table 3: Uncertainty Scores for the Amazon Dataset (5 Categories)

without uncertainty measurement concerns.

For the binary classification task, we use the accuracy and F1-score to measure the classification performance based on the testing set $S \setminus S_r$ for different uncertainty ratios r. Similarly, for multi-class tasks, we use the micro-F1 and macro-F1 scores utilizing the same settings as for the binary classification.

4.1.3 Comparison Methods

The following methods are included in the performance comparison: 1) Penultimate Layer Variance (PL-Variance). Activations before the softmax layer in a deep neural network always reveal the uncertainty of the prediction (Zaragoza and d'Alche Buc, 1998). As a baseline method, we use the variance of the output of a fully connected layer in Figure 1 as the uncertainty weight. 2) Deep Neural Networks as Gaussian Processes (NNGP) (Lee et al., 2017). This approach applies a Gaussian process to perform a Bayesian inference for deep neural networks, with a computationally efficient pipeline being used to compute the covariance function of the Gaussian process. The default parameter settings in the source code[4] were applied in our experiments. 3) Distance-based Confidence (Distance)(Mandelbaum and Weinshall, 2017). This method assigns confidence scores based on the data embedding compared to the training data. We set its nearest neighbor parameter $k = 10$. 4) Dropout (Gal and Ghahramani, 2016). Here, dropout training in DNNs is treated as an approximation of Bayesian inference in deep Gaussian processes. We set the sample number T as 100 in our experiments. 5) Dropout + Metric. In

[4] https://github.com/brain-research/nngp

		Uncertainty Ratio (Micro F1, Improved Ratio)				
		0%	**10%**	**20%**	**30%**	**40%**
Random	**DE**	0.659	0.702(6.47%)	0.748(13.46%)	0.792(20.14%)	0.831(26.03%)
	DE + Metric	0.660	**0.705(6.85%)**	**0.752(13.92%)**	**0.802(21.57%)**	**0.845(28.04%)**
Glove	**DE**	0.760	0.807(6.25%)	0.849(11.73%)	0.888(16.79%)	0.917(20.70%)
	DE + Metric	0.781	**0.835(6.93%)**	**0.878(12.47%)**	**0.918(17.62%)**	**0.944(20.92%)**

Table 4: Embedding vs. No Pre-trained Embedding

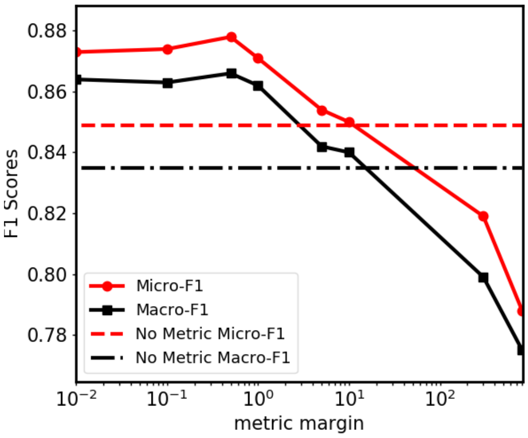


Figure 4: Prediction performance for different metric margin settings.

order to validate the effectiveness of our metric learning, we applied our proposed metric learning method to the Dropout method. The metric margin m and coefficient λ were set as 0.5 and 0.1, respectively. 6) Our proposed method. We evaluate our proposed method in two different settings, Dropout-Entropy alone (DE) and Dropout-Entropy with metric learning (DE + Metric). Here, we set the sample number $T = 100$, coefficient $\lambda = 0.1$ and the metric margin may vary from different data sets.

4.2 Experimental Results

This subsection presents the results of the uncertainty performance comparison and the analysis of the metric learning and parameter settings.

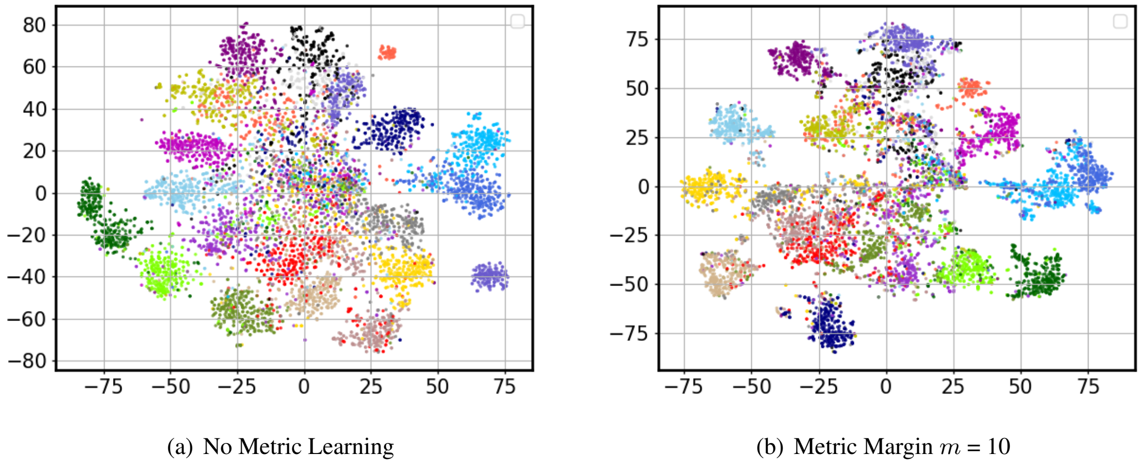
4.2.1 Uncertainty Results

Table 1 shows the Micro-F1 and Macro-F1 scores for ratios of uncertain predictions eliminated ranging from 10 to 40% for the 20NewsGroup data set. To demonstrate its effect, metric learning was also applied to the baseline method Dropout, and our proposed method DE. The improvement ratio compared to the results with no uncertainty elimination, shown in the 0% column, are presented after the F1 scores. Based on these result, we can conclude that: 1) Our proposed method, DE+Metric, significantly improves both the Micro- and Macro-F1 scores when a portion of uncertain predictions are eliminated. For example, the Micro-F1 improves from 0.78 to 0.92 when 30% of the uncertain predictions are eliminated. 2) Comparing the results obtained by DE and DE+Metric, metric learning significantly improves the results obtained for different uncertainty ratio settings. Similar results can be observed when comparing the Dropout and Dropout+Metric. For example, the Micro-F1 scores for Dropout+Metric are around 5% better than the Dropout method alone, boosting them from 0.851 to 0.892, with a 30% uncertainty ratio. 3) The DE method outperforms all the other methods when metric learning is not applied. Specifically, DE is around 4% better than the Dropout method in terms of the Micro-F1 score.

The results for IMDb and Amazon data sets are presented in Table 2 and Table 3. When comparing our proposed model's performance across three data sets, we found that the greater improvements are achieved on multi- instead of binary-class classification data sets. One possible explanation is that a comparatively large portion of multi-class features are close to the decision boundary in the feature space. Through the metric learning strategy of minimizing intra-class distance while maxmizing the inter-class instances, the feature distance between the inter-class predictions is enlarged and the quality of embeddings is greatly enhanced.

4.2.2 Analysis of Metric Learning

The impact of metric learning on feature representation is analyzed in this section. Figure 5 shows the 300-dimension feature representations



(a) No Metric Learning

(b) Metric Margin m = 10

Figure 5: Feature visualization of 20 NewsGroup testing data set in two dimensions by t-SNE algorithm.

for the 20 NewsGroup testing data set, with Figure 5(a) presenting the features trained without metric learning and Figure 5(b) those trained by metric learning with a margin parameter m=10. We used the t-SNE algorithm (Maaten and Hinton, 2008) to visualize the high dimensional features in the form of two dimensional images. From the results, we can clearly see that the distances between the inter-classes are significantly enlarged compared to the features trained without metric learning shown in Figure 5(a). This enlarged inter-class spacing means that dropout-based uncertainty methods have smaller prediction variances in case their dropout prediction trials are accurate.

4.2.3 Parameter Analysis

The impact of the metric margin and word embeddings are discussed in this section.

Metric Margin. Figure 4 shows the impact of metric margin parameters, ranging from 0 to 800 on the 20 NewsGroup data set with a 20% uncertainty ratio. From the results, we can conclude that: (1) The prediction performance is not sensitive to the point at which the metric margin parameter is set as long as its value is not extremely large. (2) Compared to the model trained with no metric learning, our methods consistently achieve better performance when the metric margin is set no larger than 10. When the metric margin is too large, however, the prediction cross-entropy loss is hard to minimize and thus dampens the overall prediction performance. (3) The results of Macro-F1 are similar to Micro-F1 with relatively small scores.

Impact of Word Embedding. We also analyzed the impact of our proposed methods on different word embedding initialization methods, including random and pre-trained Glove word vectors in 200 dimensions. Table 4 shows the results of Micro-F1 for the different uncertainty ratios. We can observe that: 1) The performance of Glove-based methods are around 15% better than that of the randomly initialized methods for different uncertainty ratios. 2) Metric learning based on a Glove initialization generally outperforms a random initialization. For instance, the F1 score of Glove rises by 0.29 when the uncertainty ratio is 20%, while for a random method it only increases by 0.04.

5 Conclusion

In this paper, a DNN-based model is proposed to address the uncertainty mitigation problem in the presence of human involvement in a text classification task. To achieve this, we proposed a dropout-entropy uncertainty measurement method with the metric learning for the feature representation. Extensive experiments on real-world data sets confirmed that our proposed approach dramatically outperforms competing methods, exhibiting a significant improvement in accuracy when a relatively small portion of the uncertainty predictions are handed over to domain experts.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

John S. Denker and Yann LeCun. 1990. Transforming neural-net output levels to probability distributions. In *Proceedings of the 3rd International Conference on Neural Information Processing Systems*, NIPS'90, pages 853–859, San Francisco, CA, USA. Morgan Kaufmann Publishers Inc.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33rd International Conference on International Conference on Machine Learning - Volume 48*, ICML'16, pages 1050–1059. JMLR.org.

Maoguo Gong, Yan Liang, Jiao Shi, Wenping Ma, and Jingjing Ma. 2013. Fuzzy c-means clustering with local information and kernel metric for image segmentation. *IEEE Transactions on Image Processing*, 22(2):573–584.

Ian J Goodfellow, Jonathon Shlens, and Christian Szegedy. 2014. Explaining and harnessing adversarial examples. *arXiv preprint arXiv:1412.6572*.

Matthieu Guillaumin, Jakob Verbeek, and Cordelia Schmid. 2009. Is that you? metric learning approaches for face identification. In *Computer Vision, 2009 IEEE 12th international conference on*, pages 498–505. IEEE.

José Miguel Hernández-Lobato and Ryan Adams. 2015. Probabilistic backpropagation for scalable learning of bayesian neural networks. In *International Conference on Machine Learning*, pages 1861–1869.

Cheng-Kang Hsieh, Longqi Yang, Yin Cui, Tsung-Yi Lin, Serge Belongie, and Deborah Estrin. 2017. Collaborative metric learning. In *Proceedings of the 26th International Conference on World Wide Web*, pages 193–201. International World Wide Web Conferences Steering Committee.

Mohit Iyyer, Jordan Boyd-Graber, Leonardo Claudino, Richard Socher, and Hal Daumé III. 2014. A neural network for factoid question answering over paragraphs. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 633–644.

Alex Kendall and Yarin Gal. 2017. What uncertainties do we need in bayesian deep learning for computer vision? In *Advances in neural information processing systems*, pages 5574–5584.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems*, pages 6405–6416.

Ken Lang. 1995. Newsweeder: Learning to filter netnews. In *Proceedings of the Twelfth International Conference on Machine Learning*, pages 331–339.

Jaehoon Lee, Yasaman Bahri, Roman Novak, Samuel S Schoenholz, Jeffrey Pennington, and Jascha Sohl-Dickstein. 2017. Deep neural networks as gaussian processes. *arXiv preprint arXiv:1711.00165*.

Andrew L. Maas, Raymond E. Daly, Peter T. Pham, Dan Huang, Andrew Y. Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 142–150, Portland, Oregon, USA. Association for Computational Linguistics.

Laurens van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605.

Amit Mandelbaum and Daphna Weinshall. 2017. Distance-based confidence score for neural network classifiers. *arXiv preprint arXiv:1709.09844*.

Rowan McAllister, Yarin Gal, Alex Kendall, Mark Van Der Wilk, Amar Shah, Roberto Cipolla, and Adrian Vivian Weller. 2017. Concrete problems for autonomous vehicle safety: Advantages of bayesian deep learning. International Joint Conferences on Artificial Intelligence, Inc.

Julian McAuley and Jure Leskovec. 2013. Hidden factors and hidden topics: understanding rating dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender systems*, pages 165–172. ACM.

Ravina Mithe, Supriya Indalkar, and Nilam Divekar. 2013. Optical character recognition. *International journal of recent technology and engineering (IJRTE)*, 2(1):72–75.

Omkar M Parkhi, Andrea Vedaldi, Andrew Zisserman, et al. 2015. Deep face recognition. In *BMVC*, volume 1, page 6.

Jeffrey Pennington, Richard Socher, and Christopher D. Manning. 2014. Glove: Global vectors for word representation. In *Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543.

Carl Edward Rasmussen. 2004. Gaussian processes in machine learning. In *Advanced lectures on machine learning*, pages 63–71. Springer.

Glenn Shafer and Vladimir Vovk. 2008. A tutorial on conformal prediction. *Journal of Machine Learning Research*, 9(Mar):371–421.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Volodya Vovk, Alexander Gammerman, and Craig Saunders. 1999. Machine-learning applications of algorithmic randomness.

Hao Wang and Dit-Yan Yeung. 2016. Towards bayesian deep learning: A survey. *arXiv preprint arXiv:1604.01662*.

Kilian Q Weinberger, John Blitzer, and Lawrence K Saul. 2006. Distance metric learning for large margin nearest neighbor classification. In *Advances in neural information processing systems*, pages 1473–1480.

Jos van der Westhuizen and Joan Lasenby. 2017. Bayesian lstms in medicine. *arXiv preprint arXiv:1706.01242*.

Eric P Xing, Michael I Jordan, Stuart J Russell, and Andrew Y Ng. 2003. Distance metric learning with application to clustering with side-information. In *Advances in neural information processing systems*, pages 521–528.

Zhixiang (Eddie) Xu, Minmin Chen, Kilian Q. Weinberger, and Fei Sha. 2012. From sbow to dcot marginalized encoders for text representation. In *Proceedings of the 21st ACM International Conference on Information and Knowledge Management*, CIKM '12, pages 1879–1884, New York, NY, USA. ACM.

Zichao Yang, Diyi Yang, Chris Dyer, Xiaodong He, Alex Smola, and Eduard Hovy. 2016. Hierarchical attention networks for document classification. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1480–1489.

Hugo Zaragoza and Florence d'Alche Buc. 1998. Confidence measures for neural network classifiers. In *Proceedings of the Seventh Int. Conf. Information Processing and Management of Uncertainty in Knowlegde Based Systems*.