Estimating the Uncertainty of Average F1 Scores

Dell Zhang
DCSIS
Birkbeck, University of London
Malet Street
London WC1E 7HX, UK
dell.z@ieee.org

Jun Wang
Dept of Computing Science
University College London
Gower Street
London WC1E 6BT, UK
j.wang@cs.ucl.ac.uk

Xiaoxue Zhao
Dept of Computing Science
University College London
Gower Street
London WC1E 6BT, UK
x.zhao@cs.ucl.ac.uk

ABSTRACT

In multi-class text classification, the performance (effectiveness) of a classifier is usually measured by micro-averaged and macro-averaged F_1 scores. However, the scores themselves do not tell us how reliable they are in terms of forecasting the classifier's future performance on unseen data. In this paper, we propose a novel approach to explicitly modelling the uncertainty of average F_1 scores through Bayesian reasoning.

Categories and Subject Descriptors

H.3.4 [**Information Storage and Retrieval**]: Systems and Software—*performance evaluation (efficiency and effectiveness)*; I.5.2 [**Pattern Recognition**]: Design Methodology—*classifier design and evaluation*

General Terms

Experimentation, Measurement, Performance

Keywords

Text Classification; Performance Evaluation; Bayesian Inference

1. INTRODUCTION

Automatic text classification [7] is a fundamental technique in information retrieval (IR) [4]. It has many important applications, including topic categorisation, spam filtering, sentiment analysis, message routing, language identification, genre detection, authorship attribution, and so on. In fact, most modern IR systems for search, recommendation, or advertising contain multiple components that use some form of text classification.

The most widely used performance measure for text classification is the F_1 score [8] which is defined as the harmonic mean of precision and recall. It is known to be more informative and more useful than classification accuracy etc. due to the prevalent phenomenon of class imbalance in text classification. When multiple classes exist in the document collection (such as Reuters-21578 with its 118 classes), we often want to compute a single aggregate measure that combines the F_1 scores for individual classes. There are two methods to do this: micro-averaging and macro-averaging [4]. The former pools per-document decisions across classes, and then computes the overall F_1 score on the pooled contingency table. The latter just computes a simple average of the F_1 scores over classes. The differences between these two averaging methods can be large: micro-averaging gives equal weight to each per-document classification decision and therefore is dominated by large classes, whereas macro-averaging gives equal weight to each class. It is nowadays a common practice for IR researchers to evaluate a multi-class text classifier using both the micro-averaged F_1 score (denoted as miF_1) and the macro-averaged F_1 score (denoted as maF_1), since their introduction by Yang and Liu's seminal SIGIR-1999 paper [9].

However, the average F_1 scores themselves only reflect a text classifier's performance on the given test data. How can we be sure that it will work well on unseen data? Given any finite amount of test results, we can never be guaranteed that one classifier's performance will definitely achieve a certain acceptable level (say 0.80) in practice. For example, suppose that a classifier got miF_1 0.81 on 100 test documents. Due to the small number of test documents, we probably do not have much confidence in pronouncing that its future performance will definitely be above 0.80. If instead the classifier got miF_1 0.81 on 100,000 test documents, we can be more confident than in the previous case. Nevertheless, there will always be some degree of uncertainty. The central question here is how to assess the uncertainty of a classifier's performance as measured by miF_1 and maF_1, given a set of test results.

In this paper, we address this problem by appealing to Bayesian reasoning [3], and demonstrate that our approach provides rich information about a multi-class text classifiers' performance.

2. OUR APPROACH

2.1 Model

Let us consider a multi-class classifier which has been tested on a collection of N labelled test documents, $\mathcal{D}$. Here we focus on the setting of multi-class single-label (aka "one-of") classification where one document belongs to one and only one class [4, 7]. For each document $\boldsymbol{x}_i$ $(i = 1, \ldots, N)$,

ICTIR'15, September 27–30, Northampton, MA, USA.
 ISBN 978-1-4503-3833-2/15/09 ...$15.00.
DOI: http://dx.doi.org/10.1145/2808194.2809488.

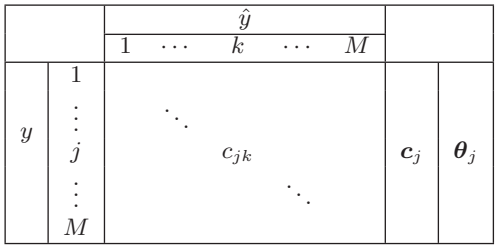


Figure 1: A schematic diagram of confusion matrix.

we have its true class label y_i as well as its predicted class label $\hat{y}_i$. Given that there are M different classes, the classification results could be fully summarised into an $M \times M$ confusion matrix $\boldsymbol{C}$ where the element c_{jk} at the j-th row and the k-th column represents the number of documents with true class label j but predicted class label k, as shown in Figure 1.

The performance measures miF_1 and maF_1 can be calculated straightforwardly based on such a confusion matrix. However, as we have explained earlier, we are not satisfied with knowing only a single score value of the performance measure, but instead would like to treat the performance measure (either miF_1 or maF_1) as a random variable ψ and estimate its uncertainty by examining its posterior probability distribution.

The test documents can be considered as "independent trials", i.e., their true class labels y_i are independent and identically distributed (i.i.d.). For each test document, we use $\boldsymbol{\mu} = (\mu_1, \ldots, \mu_M)$ to represent the probabilities that it truly belongs to each class: $\mu_j = \Pr[y_i = j]$ $(j = 1, \ldots, M)$, $\sum_{j=1}^{M} \mu_j = 1$. This means that the class sizes $\boldsymbol{n} = (n_1, \ldots, n_M)$ would follow a Multinomial distribution with parameter N and $\boldsymbol{\mu}$: $\boldsymbol{n} \sim \text{Mult}(N, \boldsymbol{\mu})$, i.e.,

$$\begin{aligned}\Pr[\boldsymbol{n}|N, \boldsymbol{\mu}] &= \frac{N!}{n_1! \ldots n_M!} \prod_{j=1}^{M} \mu_j^{n_j} \\ &= \frac{\Gamma\left(\sum_{j=1}^{M}(n_j+1)\right)}{\prod_{j=1}^{M}\Gamma(n_j+1)} \prod_{j=1}^{M} \mu_j^{n_j} \ .\end{aligned}$$

It would then be convenient to use the Dirichlet distribution (which is conjugate to the Multinomial distribution) as the prior distribution of parameter $\boldsymbol{\mu}$. More specifically, $\boldsymbol{\mu} \sim \text{Dir}(\boldsymbol{\beta})$, i.e.,

$$\Pr[\boldsymbol{\mu}] = \frac{\Gamma\left(\sum_{j=1}^{M}\beta_j\right)}{\prod_{j=1}^{M}\Gamma(\beta_j)} \prod_{j=1}^{M} \mu_j^{\beta_j - 1} \ ,$$

where the hyper-parameter $\boldsymbol{\beta} = (\beta_1, \ldots, \beta_M)$ encodes our prior belief about each class's proportion in the test document collection. If we do not have any prior knowledge, we can simply set $\boldsymbol{\beta} = (1, \ldots, 1)$ that yields a uniform distribution, as we did in our experiments.

Furthermore, let $\boldsymbol{c}_j = (c_{j1}, \ldots, c_{jM})$ denote the j-th row of the confusion matrix. In other words, $\boldsymbol{c}_j$ shows how those documents belonging to class j are classified. For each test document from that class j, we use $\boldsymbol{\theta}_j = (\theta_{j1}, \ldots, \theta_{jM})$ to represent the probabilities that it is classified into different classes: $\theta_{jk} = \Pr[\hat{y}_i = k | y_i = j]$ $(k = 1, \ldots, M)$, $\sum_{k=1}^{M} \theta_{jk} = 1$. This means that for each class j, the corresponding vector $\boldsymbol{c}_j$ would follow a Multinomial distribution with parameter n_j and $\boldsymbol{\theta}_j$: $\boldsymbol{c}_j \sim \text{Mult}(n_j, \boldsymbol{\theta}_j)$, i.e.,

$$\begin{aligned}\Pr[\boldsymbol{c}_j|n_j, \boldsymbol{\theta}_j] &= \frac{n_j!}{c_{j1}! \ldots c_{jM}!} \prod_{k=1}^{M} \theta_{jk}^{c_{jk}} \\ &= \frac{\Gamma\left(\sum_{k=1}^{M}(c_{jk}+1)\right)}{\prod_{k=1}^{M}\Gamma(c_{jk}+1)} \prod_{k=1}^{M} \theta_{jk}^{c_{jk}} \ .\end{aligned}$$

It would then be convenient to use the Dirichlet distribution (which is conjugate to the Multinomial distribution) as the prior distribution of parameter $\boldsymbol{\theta}_j$. More specifically, $\boldsymbol{\theta}_j \sim \text{Dir}(\boldsymbol{\alpha}_j)$, i.e.,

$$\Pr[\boldsymbol{\theta}_j] = \frac{\Gamma\left(\sum_{k=1}^{M}\alpha_{jk}\right)}{\prod_{k=1}^{M}\Gamma(\alpha_{jk})} \prod_{k=1}^{M} \theta_{jk}^{\alpha_{jk}-1} \ ,$$

where the hyper-parameter $\boldsymbol{\alpha}_j = (\alpha_{j1}, \ldots, \alpha_{jM})$ encodes our prior belief about a classifier's prediction accuracy for class j. If we do not have any prior knowledge, we can simply set for each class $\boldsymbol{\alpha}_j = (1, \ldots, 1)$ that yields a uniform distribution, as we did in our experiments.

Once the parameters $\boldsymbol{\mu}$ and $\boldsymbol{\theta}_j$ $(j = 1, \ldots, M)$ have been estimated, it will be easy to calculate, for each class, the contingency table of "expected" prediction results: true positive (tp), false positive (fp), true negative (tn), and false negative (fn). For example, the anticipated number of true positive predictions, for class j, should be the number of test documents belonging to that class $N\mu_j$ times the rate of being predicted by the classifier into that class as well θ_{jj}. The equations to calculate the contingency table for each class j are listed as follows.

$$\begin{aligned} tp_j &= N\mu_j\theta_{jj} & fp_j &= \sum_{u\neq j} N\mu_u\theta_{uj} \\ fn_j &= \sum_{v\neq j} N\mu_j\theta_{jv} & tn_j &= \sum_{u\neq j}\sum_{v\neq j} N\mu_u\theta_{uv} \end{aligned}$$

In *micro-averaging*, we pool the per-document predictions across classes, and then use the pooled contingency table to compute the micro-averaged precision P, micro-averaged recall R, and finally their harmonic mean miF_1 as follows.

$$\begin{aligned} P &= \frac{\sum_{j=1}^{M} tp_j}{\sum_{j=1}^{M}(tp_j + fp_j)} = \sum_{j=1}^{M} \mu_j\theta_{jj} \\ R &= \frac{\sum_{j=1}^{M} tp_j}{\sum_{j=1}^{M}(tp_j + fn_j)} = \sum_{j=1}^{M} \mu_j\theta_{jj} \\ \text{mi}F_1 &= \frac{2PR}{P+R} = \sum_{j=1}^{M} \mu_j\theta_{jj} \end{aligned}$$

It is a well-known fact that in multi-class single-label (aka "one-of") classification, mi$F_1 = P = R$ which is actually identical to the overall accuracy of classification [4].

In *macro-averaging*, we use the contingency table of each individual class j to compute that particular class's precision P_j as well as recall R_j, and finally compute a simple average of the F_1 scores over classes to get maF_1 as follows.

$$\begin{aligned} P_j &= \frac{tp_j}{tp_j + fp_j} = \frac{\mu_j\theta_{jj}}{\sum_{u=1}^{M}\mu_u\theta_{uj}} \\ R_j &= \frac{tp_j}{tp_j + fn_j} = \theta_{jj} \end{aligned}$$

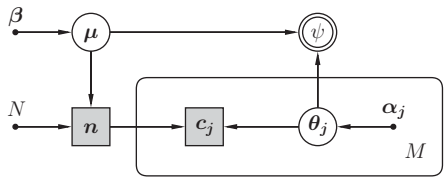


Figure 2: The probabilistic graphical model for estimating the uncertainty of average F_1 scores.

$$\text{ma}F_1 = \left(\sum_{j=1}^{M} \frac{2P_j R_j}{P_j + R_j} \right) / M$$

In the above calculation of miF_1 and maF_1, N has been cancelled out so it does not appear in the final formulae. Therefore the deterministic variable ψ for the performance measure of interest (either miF_1 or maF_1) is a function that depends on $\boldsymbol{\mu}$ and $\boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M$ only:

$$\psi = f(\boldsymbol{\mu}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M) \ .$$

The above model describes the generative mechanism of a multi-class classifier's test results (i.e., confusion matrix). It is summarised as follows, and also depicted in Figure 2 as a probabilistic graphical model (PGM) [2] using common notations.

$$\begin{aligned}
\boldsymbol{\mu} &\sim \text{Dir}(\boldsymbol{\beta}) \\
\boldsymbol{n} &\sim \text{Mult}(N, \boldsymbol{\mu}) \\
\boldsymbol{\theta}_j &\sim \text{Dir}(\boldsymbol{\alpha}_j) \qquad \text{for } j = 1, \ldots, M \\
\boldsymbol{c}_j &\sim \text{Mult}(n_j, \boldsymbol{\theta}_j) \text{ for } j = 1, \ldots, M \\
\psi &= f(\boldsymbol{\mu}, \boldsymbol{\theta}_1, \ldots, \boldsymbol{\theta}_M)
\end{aligned}$$

Our model can be considered as a generalisation of the two-class F_1 score model proposed by Goutte and Gaussier [1] to multiple classes. More importantly, it opens up many possibilities for adaptation or extension.

2.2 Implementation

The purpose of building the above model for classification results is to assess the Bayesian posterior probability of ψ that represents either miF_1 or maF_1. An approximate estimation of ψ can be obtained by sampling from its posterior probability distribution via Markov Chain Monte Carlo (MCMC) [3] techniques.

We have implemented our model with an MCMC method *Metropolis-Hastings sampling* [3]. The default configuration is to generate 50,000 samples, with no "burn-in", "lag", or "multiple-chains". The program is written in Python utilising the module `PyMC3`[1] [5] for MCMC based Bayesian model fitting. The source code will be made open to the research community on the first author's homepage.

3. EXPERIMENTS

In order to demonstrate the usage of our model for estimating the uncertainty of average F_1 scores, we have conducted experiments on the confusion matrix given by the test results from a multi-class classifier on a real-world text dataset. The confusion matrix provides all the data that our model requires. It is shown in Figure 3 to ensure the reproducibility of experiments.

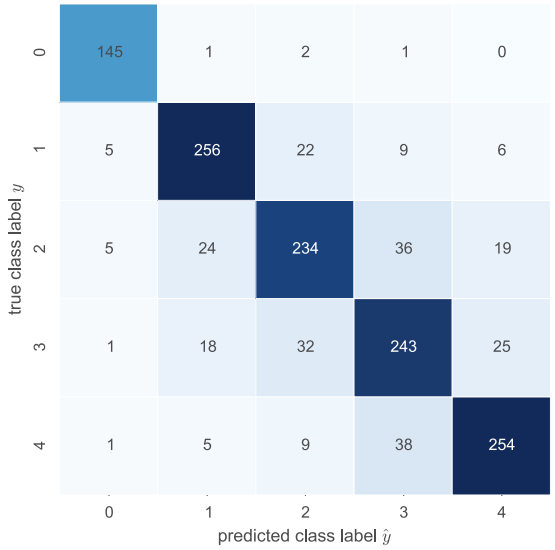


Figure 3: The confusion matrix used for our experiments.

Our proposed Bayesian estimation approach offers rich information about the given classifier's average F_1 scores, as shown in Table 1. In addition to the original performance score (miF_1 or maF_1), we have shown its posterior mean, standard deviation (std), Monte Carlo error (MC error), the percentage lower or greater than the reference performance score 0.8 (LG pct), and the 95% Highest Density Interval (HDI). In particular, the 95% HDI is a useful summary of where the bulk of the most credible values of ψ falls: by definition, every value inside the HDI has higher probability density than any value outside the HDI, and the total mass of points inside the 95% HDI is 95% of the distribution [3].

The Bayesian estimations of miF_1 and maF_1 are visualised in Figure 4 and 5 respectively. The left component in each figure plots the posterior probability distribution of the performance measure variable ψ, while the right component plots the corresponding MCMC trace which proves the convergence of sampling.

4. CONCLUSIONS

The main contribution of this paper is a Bayesian estimation approach to assessing the uncertainty of average F_1 scores in the context of multi-class text classification. Obviously the more general F_β measure ($\beta \geq 0$) [4, 8] can be dealt with in the same way.

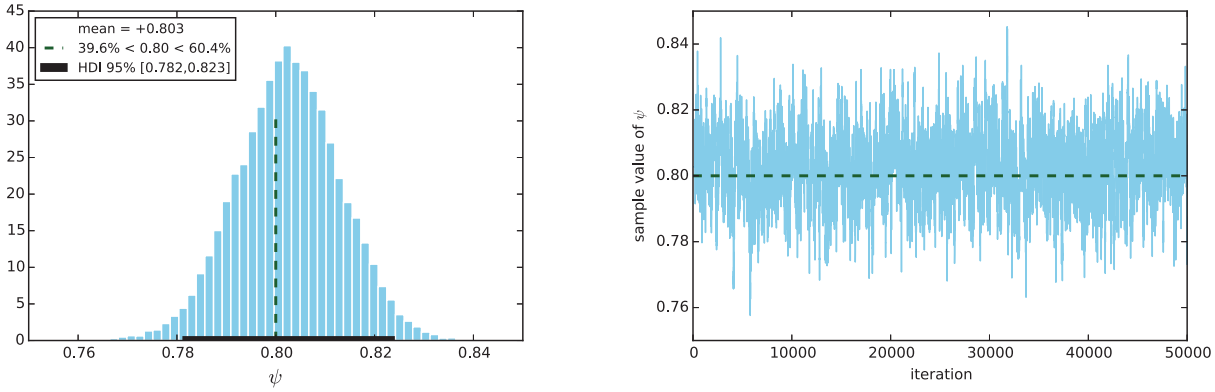
Our model for estimating the uncertainty of average F_1 scores has been described in the multi-class single-label (aka "one-of") classification setting, but it is readily extensible to the multi-class multi-label (aka "any-of") classification setting [4, 7]. In that case, the Dirichlet/Multinomial distributions should simply be replaced by multiple Beta/Binomial distributions each of which corresponds to one specific target class, because a multi-class multi-label classifier is nothing more than a composition of independent binary classifiers.

By modelling the full posterior probability distribution of miF_1 or maF_1, we are able to make meaningful *interval estimation* (e.g., the 95% HDI) instead of simplistic *point*

[1] `http://pymc-devs.github.io/pymc3/`

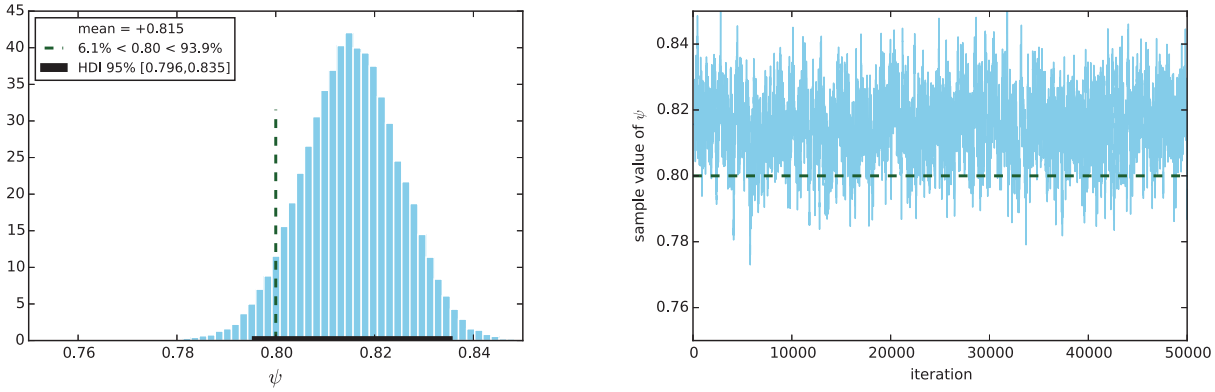
Table 1: Bayesian estimation of the average F_1 scores.

	score	mean	std	MC error	LG pct	HDI
miF_1	0.814	0.803	0.011	0.000	39.6%<0.8<60.4%	$[0.782, 0.823]$
maF_1	0.828	0.815	0.010	0.000	6.1%<0.8<93.9%	$[0.796, 0.835]$



(a) posterior plot (b) trace plot

Figure 4: Bayesian estimation of miF_1.



(a) posterior plot (b) trace plot

Figure 5: Bayesian estimation of maF_1.

estimation of a text classifier's future performance on unseen data. The rich information provided by our model will allow us to make comprehensive performance comparisons between text classifiers, by taking the uncertainty of average F_1 scores into account. It would be interesting to conduct more extensive experiments to verify whether the proposed Bayesian approach has advantages over traditional hypothesis testing [6, 9].

5. REFERENCES

[1] C. Goutte and E. Gaussier. A probabilistic interpretation of precision, recall and F-score, with implication for evaluation. In *Proceedings of the 27th European Conference on IR Research (ECIR)*, pages 345–359, Santiago de Compostela, Spain, 2005.

[2] D. Koller and N. Friedman. *Probabilistic Graphical Models - Principles and Techniques*. MIT Press, 2009.

[3] J. K. Kruschke. *Doing Bayesian Data Analysis: A Tutorial with R, JAGS, and Stan*. Academic Press, 2nd edition, 2014.

[4] C. D. Manning, P. Raghavan, and H. Schütze. *Introduction to Information Retrieval*. Cambridge University Press, 2008.

[5] A. Patil, D. Huard, and C. J. Fonnesbeck. PyMC: Bayesian stochastic modelling in Python. *Journal of Statistical Software*, 35(4):1–81, 2010.

[6] T. Sakai. Evaluating evaluation metrics based on the bootstrap. In *Proceedings of the 29th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 525–532, Seattle, WA, USA, 2006.

[7] F. Sebastiani. Machine learning in automated text categorization. *ACM Computing Surveys (CSUR)*, 34(1):1–47, 2002.

[8] C. J. van Rijsbergen. *Information Retrieval*. Butterworths, London, UK, 2nd edition, 1979.

[9] Y. Yang and X. Liu. A re-examination of text categorization methods. In *Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR)*, pages 42–49, Berkeley, CA, USA, 1999.