F-Measure Maximization in Topical Classification

Weiwei Cheng[1], Krzysztof Dembczyński[2], Eyke Hüllermeier[1], Adrian Jaroszewicz[2], and Willem Waegeman[3]

[1] Department of Mathematics and Computer Science, Marburg University, Germany
{cheng,eyke}@mathematik.uni-marburg.de
[2] Institute of Computing Science, Poznań University of Technology, Poland
{kdembczynski,ajaroszewicz}@cs.put.poznan.pl
[3] Department of Mathematical Modelling, Statistics and Bioinformatics, Ghent University, Belgium
willem.waegeman@ugent.be

Abstract. The F-measure, originally introduced in information retrieval, is nowadays routinely used as a performance metric for problems such as binary classification, multi-label classification, and structured output prediction. In this paper, we describe our methods applied in the JRS 2012 Data Mining Competition for topical classification, where the instance-based F-measure is used as the evaluation metric. Optimizing such a measure is a statistically and computationally challenging problem, since no closed-form maximizer exists. However, it has been shown recently that the F-measure maximizer can be efficiently computed if some properties of the label distribution are known. For independent labels, it is enough to know marginal probabilities. An algorithm based on dynamic programming is then able to compute the F-measure maximizer in cubic time with respect to the number of labels. For dependent labels, one needs a quadratic number (with respect to the number of labels) of parameters for the joint distribution to compute (also in cubic time) the F-measure maximizer. These results suggest a two step procedure. First, an algorithm estimating the required parameters of the distribution has to be run. Then, the inference algorithm computing the F-measure maximizer is used over these estimates. Such a procedure achieved a very satisfactory result in the JRS 2012 Data Mining Competition.

1 Introduction

While being rooted in information retrieval [1], the so-called F-measure is nowadays routinely used as a performance metric for different types of prediction problems, including binary classification, multi-label classification (MLC), and certain applications of structured output prediction, like text chunking and named entity recognition. Compared to measures like the 0-1 loss in binary classification and the Hamming loss in MLC, it enforces a better balance between performance on the minority and the majority class, and it is hence more suitable in the case of imbalanced data, which arises quite frequently in real-world applications.

J.T. Yao et al. (Eds.): RSCTC 2012, LNAI 7413, pp. 439–446, 2012.

The predictive task in the JRS 2012 Data Mining Competition[1] falls into such a category. Generally speaking, this competition concerns the topical classification of biomedical research papers based on the concept information from the MeSH ontology,[2] which are automatically assigned by the tagging system. More precisely, as the training data, there are in total 10000 instances with 25640 features and 83 classes. The values of the features are presented as integers ranging from 0 to 1000, expressing association strengths to corresponding MeSH terms, and the classes correspond to the topic identifiers. There are another 10000 instances as the test data. They share the same format as the training data, except that the class information is not given. Similar to other text classification problems, the data of the JRS competition are very sparse. Consider the training data for example, the most dense feature has 2738 nonzero entries and the most dense class is associated with 2475 instances. The sparseness of the data calls for evaluation metrics like the F-measure. More precisely, the instance-based F-measure is applied in the JRS competition, which we shall discuss later in more details.

The paper is organized as follows. We first introduce the formal setting of multi-label classification and the definition of the instance-based F-measure in Section 2. Inference techniques for F-measure maximization are discussed in Section 3, where we start with the case of independent class labels and then discuss the more general case without the independence assumption. These inference techniques are based on the parameters of the label distribution. We discuss the estimation of such parameters in Section 4. Some empirical evaluations of our approaches are shown in Section 5, prior to the final conclusion in Section 6.

2 Multi-label Learning and Instance-Based F-Measure

The task of the JRS competition is a multi-label learning problem. Let $\mathcal{X}$ denote an instance space, and let $\mathcal{L} = \{\lambda_1, \lambda_2, \ldots, \lambda_m\}$ be a finite set of class labels. An instance $\boldsymbol{x} \in \mathcal{X}$ is (non-deterministically) associated with a subset of labels $L \in 2^{\mathcal{L}}$; this subset is called the set of relevant labels, while the complement $\mathcal{L} \setminus L$ is considered as irrelevant for $\boldsymbol{x}$. It is common to identify L with a binary vector $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)$, where $y_i = 1$ means $\lambda_i \in L$. We denote the set of possible labelings as $\mathcal{Y} = \{0, 1\}^m$.

Given a prediction $\boldsymbol{h}(\boldsymbol{x}) = (h_1(\boldsymbol{x}), \ldots, h_m(\boldsymbol{x})) \in \mathcal{Y}$ of an m-dimensional binary label vector $\boldsymbol{y} = (y_1, \ldots, y_m)$, the label vector associated with a single instance, the instance-based F-measure is defined as follows:

$$F(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{2 \sum_{i=1}^{m} y_i h_i(\boldsymbol{x})}{\sum_{i=1}^{m} y_i + \sum_{i=1}^{m} h_i(\boldsymbol{x})} \in [0, 1] \,, \tag{1}$$

[1] http://tunedit.org/challenge/JRS12Contest

[2] http://www.nlm.nih.gov/mesh/introduction.html

where $0/0 = 1$ by definition. This measure essentially corresponds to the harmonic mean of precision *prec* and recall *rec*:

$$prec(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{\sum_{i=1}^{m} y_i h_i(\boldsymbol{x})}{\sum_{i=1}^{m} h_i(\boldsymbol{x})}, \quad rec(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) = \frac{\sum_{i=1}^{m} y_i h_i(\boldsymbol{x})}{\sum_{i=1}^{m} y_i} \,. \tag{2}$$

One can generalize the F-measure to a weighted harmonic average of these two values, but for the sake of simplicity, we stick to the unweighted mean, which is often referred to as the F1-score or the F1-measure. This variant of the F-measure was also used in the competition.

Modeling the ground-truth as a random variable $\boldsymbol{Y}$, i.e., assuming an underlying probability distribution $p(\boldsymbol{Y})$ on $\{0,1\}^m$, the prediction $\boldsymbol{h}_F^*(\boldsymbol{x})$ that maximizes the expected F-measure is given by

$$\begin{aligned} \boldsymbol{h}_F^*(\boldsymbol{x}) &= \operatorname*{arg\,max}_{\boldsymbol{h}(\boldsymbol{x}) \in \{0,1\}^m} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{Y})} \left[F(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \right] \\ &= \operatorname*{arg\,max}_{\boldsymbol{h}(\boldsymbol{x}) \in \{0,1\}^m} \sum_{\boldsymbol{y} \in \{0,1\}^m} p(\boldsymbol{Y}{=}\boldsymbol{y})\, F(\boldsymbol{y}, \boldsymbol{h}(\boldsymbol{x})) \,. \end{aligned} \tag{3}$$

Unfortunately, a closed form of the maximizer $\boldsymbol{h}_F^*(\boldsymbol{x})$ does not exist and a brute-force search is infeasible, as it would require checking all 2^m combinations of prediction vector $\boldsymbol{h}$ and computing a sum over an exponential number of terms for each $\boldsymbol{h}$. However, several algorithms have been introduced recently that compute the F-measure maximizer efficiently.

3 Algorithms for F-Measure Maximization

The problem (3) can be solved via outer and inner maximization [2]. Namely, (3) can be transformed into an inner maximization

$$\boldsymbol{h}^{(k)^*} = \operatorname*{arg\,max}_{\boldsymbol{h} \in H_k} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{Y})} \left[F(\boldsymbol{y}, \boldsymbol{h}) \right] \,, \tag{4}$$

where $H_k = \{\boldsymbol{h} \in \{0,1\}^m \mid \sum_{i=1}^m h_i = k\}$, followed by an outer maximization

$$\boldsymbol{h}_F^* = \operatorname*{arg\,max}_{\boldsymbol{h} \in \{\boldsymbol{h}^{(0)^*}, \ldots, \boldsymbol{h}^{(m)^*}\}} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{Y})} \left[F(\boldsymbol{y}, \boldsymbol{h}) \right] \,. \tag{5}$$

The outer maximization (5) can be done by simply checking all $m{+}1$ possibilities. The main effort is then required for solving the inner maximization (4).

3.1 Label Independence

By assuming independence of the random variables $Y_1, \ldots, Y_m$, the optimization problem (3) can be substantially simplified. It has been shown independently in [3] and [2] that the optimal solution always contains the labels with the highest marginal probabilities $p_i = P(Y_i = 1)$, or no labels at all. As a consequence, only a few ($m + 1$ instead of 2^m) hypotheses $\boldsymbol{h}$ need to be examined.

Furthermore, Lewis [3] has shown that the expected F-measure can be approximated by the following expression under the assumption of independence:[3]

$$\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{Y})}\left[F(\boldsymbol{y},\boldsymbol{h})\right] \simeq \begin{cases} \prod_{i=1}^{m}(1-p_i) & \text{if } \boldsymbol{h}=\boldsymbol{0}\ , \\ \dfrac{2\sum_{i=1}^{m} p_i h_i}{\sum_{i=1}^{m} p_i + \sum_{i=1}^{m} h_i} & \text{if } \boldsymbol{h}\neq\boldsymbol{0}\ . \end{cases} \tag{6}$$

This approximation is exact for $\boldsymbol{h}=\boldsymbol{0}$, and is tractable with $\mathcal{O}(m)$. For $\boldsymbol{h}\neq\boldsymbol{0}$, an upper bound of the error can easily be determined [3]. However, the exact solution can be computed efficiently, as will be explained in more details below.

Jansche [2] and Chai [4] have independently proposed exact procedures for solving the inner maximization (4). The former runs in $\mathcal{O}(m^3)$, while the latter runs in $\mathcal{O}(m^2)$, leading to the overall complexity of $\mathcal{O}(m^4)$ and $\mathcal{O}(m^3)$, respectively. Since both algorithms deliver the same estimate, we focus on Chai's approach here. We refer to it as DP, since it is based on dynamic programming.

Chai [4] has shown that the expected F-measure of $\boldsymbol{h}^{(k)^*}$, the solution of the inner maximization (4) for a given k that assigns ones to k labels with the largest marginal probabilities, can be expressed as follows:

$$\mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{Y})}\left[F(\boldsymbol{y},\boldsymbol{h}^{(k)})\right] = 2\prod_{i=1}^{m}(1-p_i)I_1(m)\ ,$$

where $I_1(m)$ is given by the following recurrent equations and boundary conditions:

$$\begin{aligned} I_t(a) &= I_{t+1}(a) + r_t I_{t+1}(a+1) + r_t J_{t+1}(a+1) \\ J_t(a) &= J_{t+1}(a) + r_t J_{t+1}(a+1) \\ I_{k+1}(a) &= 0 \quad J_{m+1}(a) = a^{-1} \end{aligned}$$

with $r_i = p_i/(1-p_i)$. These equations suggest a dynamic programming algorithm of space $\mathcal{O}(m)$ and time $\mathcal{O}(m^2)$ for solving the inner maximization (4) for given k.

3.2 A General Procedure

If the independence assumption is violated, the above methods may produce predictions far away from the optimal one, as shown in [5] by Dembczynski et al. In this paper, the authors have further introduced an exact and efficient algorithm for computing the F-measure maximizer without using any additional assumption on the probability distribution $p(\boldsymbol{Y})$. The algorithm, called general F-measure maximizer (GFM), needs m^2+1 parameters and runs in $\mathcal{O}(m^3)$.

The inner optimization problem (4) can be formulated as follows:

$$\boldsymbol{h}^{(k)^*} = \arg\max_{\boldsymbol{h}\in H_k} \mathbb{E}_{\boldsymbol{y}\sim p(\boldsymbol{Y})}\left[F(\boldsymbol{y},\boldsymbol{h})\right] = \arg\max_{\boldsymbol{h}\in H_k} \sum_{\boldsymbol{y}\in\{0,1\}^m} p(\boldsymbol{y})\frac{2\sum_{i=1}^{m} y_i h_i}{s_{\boldsymbol{y}}+k}\ ,$$

[3] We henceforth denote $\boldsymbol{0}$ and $\boldsymbol{1}$ as vectors containing all zeros and ones, respectively.

with $s_{\boldsymbol{y}} = \sum_{i=1}^{m} y_i$. The sums can be swapped, resulting in

$$\boldsymbol{h}^{(k)^*} = \operatorname*{arg\,max}_{\boldsymbol{h} \in H_k} \; 2 \sum_{i=1}^{m} h_i \sum_{\boldsymbol{y} \in \{0,1\}^m} \frac{p(\boldsymbol{y}) y_i}{s_{\boldsymbol{y}} + k} \; . \tag{7}$$

Furthermore, one can sum up the probabilities $p(\boldsymbol{y})$ for all $\boldsymbol{y}$ with an equal value of $s_{\boldsymbol{y}}$. By using

$$p_{is} = \sum_{\boldsymbol{y} \in \{0,1\}^m : s_{\boldsymbol{y}} = s} y_i p(\boldsymbol{y}) \, ,$$

one can transform (7) into the following expression:

$$\boldsymbol{h}^{(k)^*} = \operatorname*{arg\,max}_{\boldsymbol{h} \in H_k} \; 2 \sum_{i=1}^{m} h_i \sum_{s=1}^{m} \frac{p_{is}}{s + k} \; . \tag{8}$$

As a result, one does not need the whole distribution to solve (4), but only the values of p_{is}, which can be given in the form of an $m \times m$ matrix P with entries p_{is}. For the special case of $k = 0$, we have $\boldsymbol{h}^{(k)^*} = \boldsymbol{0}$ and $\mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{Y})} \left[F(\boldsymbol{y}, \boldsymbol{0}) \right] = p(\boldsymbol{Y} = \boldsymbol{0})$.

If the matrix P and $p(\boldsymbol{Y} = \boldsymbol{0})$ are given, the solution of (3) is straight-forward. To simplify the notation, let us introduce an $m \times m$ matrix W with elements

$$w_{sk} = \frac{1}{s + k} \; , \qquad s, k \in \{1, \ldots, m\} \; . \tag{9}$$

The resulting algorithm needs then to compute the following matrix:

$$\mathrm{F} = \mathrm{PW} \; ,$$

with entries denoted by f_{ik}. The inner optimization problem (4) can then be reformulated as follows:

$$\boldsymbol{h}^{(k)^*} = \operatorname*{arg\,max}_{\boldsymbol{h} \in H_k} \; 2 \sum_{i=1}^{m} h_i f_{ik} \, .$$

The solution for a given $k \in \{1, \ldots, m\}$ is obtained by setting $h_i{=}1$ for the top k largest elements in the k-th column of the matrix F, and $h_i{=}0$ for the rest. The corresponding value of the expected F-measure for $\boldsymbol{h}^{(k)^*}$ has to be stored for being used in the outer maximization. We also need to compute a case in which $k = 0$:

$$\mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{Y})} \left[F(\boldsymbol{y}, \boldsymbol{0}) \right] = p(\boldsymbol{Y} = \boldsymbol{0}) \; .$$

The last step relies on solving the outer maximization (5):

$$\boldsymbol{h}_F^* = \operatorname*{arg\,max}_{\boldsymbol{h} \in \{\boldsymbol{h}^{(0)^*}, \ldots, \boldsymbol{h}^{(m)^*}\}} \mathbb{E}_{\boldsymbol{y} \sim p(\boldsymbol{Y})} \left[F(\boldsymbol{y}, \boldsymbol{h}) \right] \; .$$

The complexity of the above algorithm is dominated by the matrix multiplication PW that is solved naively in $\mathcal{O}(m^3)$. The algorithm needs $m^2 + 1$ parameters in total, namely the matrix P and probability $p(\boldsymbol{Y} = \boldsymbol{0})$.

3.3 Discussion

The DP approach described in Section 3.1 and GFM are characterized by a similar computational complexity, however, the former does not deliver an exact F-measure maximizer if the assumption of independence is violated. On the other hand, the DP approach relies on a smaller number of parameters (m values representing marginal probabilities). GFM needs $m^2 + 1$ parameters, but then computes the maximizer exactly. Since estimating a larger number of parameters is statistically more difficult, it is a priori unclear which method performs better in practice. We are facing here a common trade-off between an approximate method on better estimates (we need to estimate a smaller number of parameters from a given sample) and an exact method on potentially weaker estimates.

4 Learning Parameters of the Distribution

In the above section, we described two inference techniques that compute the F-measure maximizers based on delivered parameters of the label distribution. To estimate these parameters we used two well-known methods for multi-label classification: binary relevance and probabilistic classifier chains.

4.1 Binary Relevance

BR is the simplest approach to multi-label classification. It reduces the problem to binary classification, by training a separate binary classifier $h_i(\cdot)$ for each label. Learning is performed independently for each label, ignoring all other labels. Obviously, BR does not take label dependence into account, but with a proper base classifier it is able to deliver accurate estimates of marginal probabilities. These estimates can be further used as inputs in the DP inference algorithm. BR is, however, not appropriate for GFM.

4.2 PCC

PCC [6] is an approach similar to Conditional Random Fields (CRFs) [7,8], which estimates the joint conditional distribution $p(\boldsymbol{Y} \mid \boldsymbol{x})$. This approach has the additional advantage that one can easily sample from the estimated distribution. The underlying idea is to repeatedly apply the product rule of probability to the joint distribution of the labels $\boldsymbol{Y} = (Y_1, \ldots, Y_m)$:

$$p(\boldsymbol{Y} = \boldsymbol{y} \mid \boldsymbol{x}) = \prod_{i=1}^{m} p(Y_i = y_i \mid \boldsymbol{x}, y_1, \ldots, y_{i-1}) \; . \tag{10}$$

Learning in this framework can be considered as a procedure that relies on constructing probabilistic classifiers for estimating $p(Y_i = y_i | \boldsymbol{x}, y_1, \ldots, y_{i-1})$, independently for each $i = 1, \ldots, m$. By plugging the log-linear model into (10), it can be shown that pairwise dependencies between labels y_i and y_j are modeled.

To sample from the conditional joint distribution $p(\boldsymbol{Y} \mid \boldsymbol{x})$, one follows the chain and picks the value of label y_i by tossing a biased coin with probabilities given by the i-th classifier. From the sample of such observations one can estimate all the parameters required by the GFM algorithm. One can also estimate the marginal probabilities and use the DP algorithm. The result is not necessarily the same as in BR, since we are using a more complex feature space here.

5 Results in the Competition

In this section we report results on the JRS 2012 Data Mining Competition dataset of the methods we discussed in previous sections. Our preprocessing on the competition data is quite straightforward: We simply delete all the empty columns (i.e., zero vectors) in the training data, then the corresponding columns in the test data. The values of features are normalized to $[0, 1]$.

In both BR and PCC we use linear regularized logistic regression from the Mallet package[4] as a base classifier. We tune the regularization parameter for each base classifier independently by minimizing the negative log-likelihood, which should provide better probability estimates. We use 10-fold cross-validation and we choose the regularization parameter from the following set of possible values $\{10^{-5}, 10^{-4}, \ldots, 10^{5}\}$. We use PCC with both inference methods and try different sizes of sample generated from the conditional distribution of a given $\boldsymbol{x}$.

The results of the methods are presented in Table 1. The F-measure is computed over the entire test set delivered by the organizers after the competition. This is a minor difference in comparison to the competition results which are computed over 90% of test examples. The remaining 10% of test examples constitute a validation set that served for computing the scores for the leaderboard during the competition. The last row in the table gives the result of the final method we used in the competition. It relies on averaging over all predictions we computed during the competition. These predictions are the results of the approaches presented in this paper but with different parametrization. In total we gathered 16 predictions and we aggregated them via voting. In this voting procedure we tested different thresholds on the validation set and selected the best one (nine votes from 16).

From the results we can see that there is no big difference among the methods. The voting procedure improves only slightly over BR+DP and PCC+GFM. Interestingly, BR+DP performs here better than PCC+GFM, which suggests independence of the labels. However, one can also observe that PCC+DP loses against other methods. This shows that PCC with the sampling procedure has problems with the accurate estimation of the marginal probabilities. Increasing the sample size improves the results (for both, DP and GFM), but it still seems that BR+DP is the most appropriate method in this case. It is the cheapest one, since it does not require additional sampling in the inference step as PCC does, and gives results only slightly worse than the voting method that averages over many predictions.

[4] `http://mallet.cs.umass.edu/`

Table 1. The results of the presented methods obtained on the entire test set. The numbers in parentheses denote the size of the sample in PCC.

Method	F-measure	Method	F-measure
PCC+DP (50)	0.48650	PCC+GFM (50)	0.52286
PCC+DP (200)	0.51979	PCC+GFM (200)	0.53005
PCC+DP (1000)	0.52995	PCC+GFM (1000)	0.53146
BR+DP	0.53279	Voting (final submission)	0.53327

6 Conclusions

The JRS 2012 Data Mining Competition is essentially a multi-label learning problem, where the objective is to optimize the instance-based F-measure. In this paper, we have introduced several theoretically sound methods addressing this optimization problem. We have shown that, although the F-measure maximization becomes significantly simpler under the assumption of independently distributed labels, it can also be accomplished efficiently without this assumption. Our final predictions are produced by a blend of all these methods and have achieved a very satisfactory result, the second place in the competition.

Acknowledgments. Weiwei Cheng and Eyke Hüllermeier are supported by German Research Foundation (DFG). Krzysztof Dembczyński and Adrian Jaroszewicz are supported by the grant 91-515/DS funded by the Polish Ministry of Science and Higher Education. Willem Waegeman is supported as a postdoc by the Research Foundation of Flanders (FWO-Vlaanderen).

References

1. van Rijsbergen, C.J.: Foundation of evaluation. Journal of Documentation 30(4), 365–373 (1974)
2. Jansche, M.: A maximum expected utility framework for binary sequence labeling. In: ACL 2007, pp. 736–743 (2007)
3. Lewis, D.: Evaluating and optimizing autonomous text classification systems. In: SIGIR 1995, pp. 246–254 (1995)
4. Chai, A.: Expectation of F-measures: Tractable exact computation and some empirical observations of its properties. In: SIGIR 2005, pp. 593–594 (2005)
5. Dembczyński, K., Waegeman, W., Cheng, W., Hüllermeier, E.: An exact algorithm for F-measure maximization. In: NIPS 2011, 223–230 (2011)
6. Dembczyński, K., Cheng, W., Hüllermeier, E.: Bayes optimal multilabel classification via probabilistic classifier chains. In: ICML 2010, pp. 279–286 (2010)
7. Lafferty, J., McCallum, A., Pereira, F.: Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In: ICML 2001, pp. 282–289 (2001)
8. Ghamrawi, N., McCallum, A.: Collective multi-label classification. In: CIKM 2005, pp. 195–200 (2005)