Journal of Machine Learning Research 1 (2014) 1-48 Submitted 4/00; Published 10/00

Likelihood and F-measure maximization under uncertainty

Georgi Dimitroff GEORGI.DIMITROV@ONTOTEXT.COM
Semantic Annotation and Search
Ontotext
47A Tsarigradsko Shosse, Sofia 1504, Bulgaria

Trevor Rose T.ROSE@UNSW.EDU.AU
School of Mathematics and Statistics
University of New South Wales
NSW 2052, Australia

Borislav Popov BORISLAV.POPOV@ONTOTEXT.COM
Semantic Annotation and Search
Ontotext
47A Tsarigradsko Shosse, Sofia 1504, Bulgaria

Editor: ...

Abstract

It is standard to perform classification tasks under the assumption that class labels are deterministic. In this context, the F-measure is an increasingly popular measure of performance for a classifier, and expresses a flexible trade-off between precision and recall. However, it may just as easily be advisable to remove this assumption and consider instances as belonging to each class with given probabilities. The presence of uncertainty in a training set may be due to subjectivity of a classification task or noise introduced during data collection. In this paper, we adapt the classical F-measure to the uncertain context and present an efficient, easy-to-implement algorithm for the optimization of this new "noisy" F-measure within the maximum entropy modeling framework. We provide comprehensive theoretical justification along with numerical experiments that demonstrate the novelty and effectiveness of this approach.

Keywords: F-measure, precision, recall, noisy data, maximum entropy model, weighted likelihood, uncertain likelihood.

1. Introduction and motivation

It is in some situations presumptuous to think that we should have exclusively deterministic information about a given sample when training a classifier. While traditional classification procedures tend to learn only from a deterministically labeled training set, it may be the case that some of the training examples fall into each class with given probabilities, and thus follow a categorical distribution. The intuition that guides such an uncertain framework is that when humans make subjective judgements, whether deciding on a particular course of action or simply giving their opinion about a popular film or a political party, it is rarely a process of deductions from verified knowledge, but instead a process that involves uncertainty on multiple levels. Whether or not we speak from direct knowledge, it is usually belief that informs realistic decision-making which comprehensively exploits the sample available to us.

The presence of uncertainty in a training set may be due to a few possible reasons, such as the following:

(i) If the classification problem involves a high degree of subjectivity, we must contend with the bias of individuals, specifically when we have employed annotators to manually classify a sample for the purpose of training. A sentiment analysis application may be considered here, where for example if the sentiment of a social media post x_i on a particular issue is viewed as negative by 70% of polled readers and positive by 30% of them, it would make sense to assign a corresponding Bernoulli distribution to the random class $Y_i \sim B(1, 0.3)$. Since the state of objects in this problem is intrinsically random, we should not consider the output of a classification model to be deterministic either.

(ii) We consider the case where there is now an objective truth but with data having again been manually annotated. We must here deal with the aspect of human error, as explored by Cohn and Specia (2013), in order to output the best possible deterministic model classes. To do this we may consider the manually labeled examples to have noisy classifications, with the entropy of a given class governing the emphasis we place on the corresponding point in the training set when fitting our model. In this case, the less certain we are of an example's class, the less we should let it influence our future decisions.

(iii) Finally, we consider the case of semi-supervised learning, similar to Mann and McCallum (2010), where we have a great deal of data available to us, most of which has not been assigned a class. We may wish to use as much of this data as possible by providing (as rough or as precise as we like) noisy estimates of their classes. For example, we might automatically generate classes by a possibly noisy and potentially quite simple (e.g. hard-coded) rule operating on each instance. We may then use this uncertain data in the training phase. In the context of the social media example above we may assign uncertain sentiment to a post according to some specific indicators like the presence of particular smileys or keywords.

The F-measure is a useful measure of performance of a binary classifier that trades off between precision and recall (though generalizations exist for the multiclass setting). It is expressed as the weighted harmonic mean:

$$F_\beta = \left(\frac{\beta}{P} + \frac{1-\beta}{R} \right)^{-1}$$

for some $\beta \in [0, 1]$, which characterizes the trade-off, with a larger β corresponding to a higher emphasis on precision and a lower emphasis on recall. As noted by Dimitroff et al. (2013), high precision may be required in automatic summarization or machine translation problems and high recall may be required in information retrieval systems, e.g. search engines.

However, in the uncertain setting we are not presented with examples with deterministically known classes and hence we cannot apply the standard F-measure. We are motivated to extend the F-measure beyond its deterministic definition in order to assess the performance of a classifier that uses random labels. As in the deterministic case, we can associate a loss function with our predictions which provides a measure of misclassification. Practically we may wish to consider some error types as more costly than others. Thus in the familiar way it is clear that there at least should be an F-measure corresponding to this loss. Since it is not a priori clear how to appropriately generalize that F-measure to the uncertain case, and noting that there is not a unique way to do it,

we propose a natural extension with an intuitive interpretation. Our generalizations collapse to the deterministic case if the randomness in the training/test set vanishes.

Indeed, to not only assess but also improve model performance, we consider optimizing the generalized F-measure during the training phase, with a similar idea to Jansche (2005). Following Dimitroff et al. (2014), we show that we can perform uncertain F-measure optimization by maximizing a weighted uncertain likelihood while adaptively adjusting for the right weights.

Related work The still ongoing boom of big data and automatic data acquisition has sparked further interest in efficient approaches to model and mine uncertain data – see Aggarwal and Yu (2009) and Aggarwal (2009). A myriad of methods for statistical learning from such data have been developed. For example, S. Tsang and Lee (2011) adapt a decision tree technique to the noisy case. W. K. Ngai and Yip (2006) and Kriegel and Pfeifle (2005) explore clustering mining approaches to uncertain data.

The maximum likelihood paradigm is pursued by Raykar et al. (2010), who describe a probabilistic approach for supervised learning based on the interpretation of multiple annotators providing (possibly noisy) labels without an absolute gold standard. Besides giving an estimate for the actual hidden labels their approach also evaluates the experts using expectation maximization. The considered noisy likelihood function is precisely the one used in this paper. On the other hand, we have the related work by Dempster et al. (1977), where maximum likelihood methods were first applied to incomplete data. We can view uncertainty as a type of incompleteness in data, provided that we do consider there to be a deterministic gold standard. While the objective function is in general the same, we are not necessarily assuming that a deterministic (albeit not perfectly observable) gold standard exists. We rather take the distribution induced by the "panel of annotators" as a generalized ground truth and proceed with the estimation of the parameters. Multi-task Gaussian processes are used by Cohn and Specia (2013) to model annotator bias, specifically applied to machine translation. Here we have also that the idea of a deterministic ground truth is a too limiting paradigm. A further generalization of the uncertain likelihood to continuous label distributions as well as an EM-based learning algorithm is presented by Denoeux (2013).

In most of the related work it is assumed that a deterministic gold standard exists, however the observations are noisy. Manwani and Sastry (2011) investigate the noise tolerance of loss minimization learners based on different loss functions. In the recent paper by Scott et al. (2013), the authors give theoretical conditions on the type and magnitude of the noise, ensuring identifiability, and extend existing results to the non-separable case with asymmetric class-wise noise.

On the more practical side, Elkan and Noto (2008) discuss a learning procedure from only positive and unlabeled data. The missing negative examples are compensated with unlabeled ones for which a class distribution is estimated. Then the unlabeled examples are considered negative with the same estimated probability and the classifier is trained on this noisy input. Natarajan et al. (2013) follow a similar approach, however assume the noise is on both classes and is known. They suggest a weighted surrogate loss, for which they obtain strong empirical risk bounds. In this sense their approach is similar to ours as we rely on a modified weighted likelihood to account for the randomness in the training data.

Given the above cited works by Raykar et al. (2010) and Denoeux (2013) for the maximum likelihood approach to noisy data and the references therein, it seems that the uncertain likelihood is the proper generalization of the standard likelihood to the noisy case. However, for a wide variety of problems the likelihood might not be the most well suited objective function – very often one has

a problem induced loss function which assigns different loss to different misclassification cases. To evaluate a goodness of a classifier in these situations one often uses the F-measure (see e.g. Powers (2011)). We propose an appropriate generalization of the F-measure to uncertain data. We follow the approach of Dimitroff et al. (2014) and derive an efficient and straightforward-to-implement algorithm for its optimization. There is a huge literature and a variety of methods used for F-measure optimization, see e.g. Nan et al. (2012), Dembczyński et al. (2011), Jansche (2005) and the references therein, with the approach of Jansche (2005) being the closest to ours.

To our best knowledge we are not aware of any generalizations of the well known F-measure to the noisy case, nor of algorithms to optimize a precision/recall based tradeoff of an uncertain classification task.

2. The Uncertain Maximum Entropy

The maximum entropy modeling framework as introduced in the NLP domain by Berger et al. (1996) has become standard for various NLP tasks. To fix notations, consider a training set of m examples $\{(x_i, y_i) : i \in 1, \dots m\}$ where x_i's are the attributes and y_i's are the classes taking values in some finite set $\mathcal{Y} = \{c_1, c_2, \dots, c_M\}$.

Furthermore, we have a set of N features $\{f_j : j \in 1, \dots, N\}$. Each feature is a mapping:

$$f_j : \mathcal{X} \times \mathcal{Y} \ni (x, y) \mapsto f_j(x, y) \in \mathbb{R}$$

from the Cartesian product of the space of attributes $\mathcal{X}$ and the space of classes $\mathcal{Y}$ into the real numbers, typically $\{0, 1\}$. The maximum entropy principle forces the model conditional probabilities $p(y|\, x, \lambda)$, of an example with attributes x to be of class y, to be of the form:

$$p(y|\, x, \lambda) = \frac{1}{Z_\lambda(x)} \exp\left(\lambda \cdot f(x, y)\right),$$

where $\lambda \in \mathbb{R}^N$ are the parameters of the model and $Z_\lambda(x)$ is the corresponding partition function given by:

$$Z_\lambda(x) = \sum_{y \in \mathcal{Y}} \exp\left(\lambda \cdot f(x, y)\right).$$

Throughout the paper we will use this parametrization of the conditional probabilities. The calibration of the model amounts to (see Berger et al., 1996) maximizing the log-likelihood function:

$$l(\lambda : x, y) = \sum_{i=1}^{m} \log p(y_i|\, x_i, \lambda).$$

With this feature-based representation, which of course is completely equivalent to the standard logistic regression convention, it is particularly easy to make sense of the likelihood function for a training set containing examples that are not deterministically classified but rather have random labels. To stress the fact that the observed classes are random we will denote them with capital letters Y_i and the corresponding probability weights are $q_i = (q_{i1}, \dots, q_{iM})$, where $M = |\mathcal{Y}|$. For a set of training examples with random labels we define the uncertain likelihood to be:

$$L^U(\lambda : x, Y) = \prod_{i=1}^{m} \prod_{j=1}^{M} p(c_j|\, x_i, \lambda)^{q_{ij}}.$$

Correspondingly, the uncertain log-likelihood is:

$$l^U(\lambda : x, Y) = \sum_{i=1}^{m} \sum_{j=1}^{M} q_{ij} \log p(c_j | \, x_i, \lambda).$$

The interpretation is that our observations are not deterministic but random. This is the case if for example a training example has been automatically parsed and its class has been predicted with some uncertainty. To understand why the random observations Y_i lead to a likelihood function L^U, we can "emulate" the random observations by creating many copies of the observations, cloning the attributes x but assigning different classes to them according to the frequencies given by the observed distributions. If our random observation is (x_i, Y_i) with Y_i having the distribution given by the weights $(q_{i1}, \ldots, q_{iM})$, we create a large number K of observations $(x_i, y_{i1}), \ldots, (x_i, y_{iK})$, where the attributes x_i are the same and we have approximately $q_{ij} \cdot K$ copies with class c_j. If we do this for all training examples using the same large number of copies K, the corresponding likelihood function after renormalization is approximately the uncertain likelihood and in the limit as $K \to \infty$ we would obtain precisely $l^U(\lambda : x, Y)$. From this representation we know that the uncertain likelihood inherits the highly desirable properties of the standard deterministic likelihood. In particular, the log-likelihood is a concave function which can be maximized via standard algorithms, e.g. gradient ascent, with:

$$\nabla l^U = \sum_{i=1}^{m} \sum_{j=1}^{M} \frac{q_{ij}}{p(c_j | \, x_i, \lambda)} \nabla p(c_j | \, x_i, \lambda). \tag{2.1}$$

We observe also that there is an obvious extension of the uncertain log-likelihood to its weighted counterpart:

$$l^U(\lambda : w, x, Y) = \sum_{i=1}^{m} w_i \left[\sum_{j=1}^{M} q_{ij} \log p(c_j | \, x_i, \lambda) \right] = \sum_{i=1}^{m} \sum_{j=1}^{M} w_i q_{ij} \log p(c_j | \, x_i, \lambda).$$

As usual, this expression corresponds to modifying the training set by adding new examples having the same attributes and uncertain classes (x_i, Y_i) with intensity w_i.

The primary interpretation of the uncertain maximum entropy we will follow is the "panel of annotators" interpretation, where we think of the randomization of classes as being generated by a panel of annotators who have voted on the class for each example. Our job is to train a model given their votes, and since they do not always agree we end up having non-degenerate distributions over the classes, defined by their proportional votes, instead of deterministic examples. Note that this "panel of annotators" perspective is not intended to be taken literally. We aim in practice to have a number of manual annotators which is small but large enough to be informative. The panel we refer to is more likely to be made up of implicit annotations which are available en masse, and in particular, online. These implicit annotations may take the form of social media "likes", votes or the presence of various sentiment markers in text such as emoticons. Now our primary "panel of annotators" interpretation becomes also a primary motivation for considering the uncertain classification problem. In order to better harness the large amounts of data available to us online which are made up of subjective opinions of individuals who are often in disagreement, considering classes to be uncertain seems in practice very sensible.

2.1 Link to the Kullback-Leibler Divergence

The following result states that maximizing the weighted uncertain likelihood as defined in the previous section is equivalent to minimizing a weighted average Kullback-Leibler divergences between obtained model probabilities $p_i(\lambda) = p(\cdot|\, x_i, \lambda)$ and the class distributions of training examples q_i.

Proposition 2.1. *The weighted uncertain log-likelihood admits the following decomposition:*

$$l^U(\lambda : x, Y) = -\sum_{i=1}^{m} w_i \left[H(q_i) + \mathcal{D}_{KL}(q_i||p(\cdot|\, x_i, \lambda))\right] \tag{2.2}$$

where $H(q_i) = \sum_{j=1}^{M} q_{ij} \log q_{ij}$ is the entropy of the distribution of Y_i and $\mathcal{D}_{KL}(q_i||p(\cdot|\, x_i, \lambda))$ is the Kullback-Leibler divergence of the estimated class distribution $p(\cdot|\, x_i, \lambda)$ from the observed distribution of Y_i.

In particular, let $\hat{\lambda}$ be the uncertain likelihood maximizer. Then $\hat{\lambda}$ minimizes the weighted average Kullback-Leibler divergence between experimental and model distributions, that is:

$$\hat{\lambda} = \arg\min \sum_{i=1}^{m} w_i \mathcal{D}_{KL}(q_i||p(\cdot|\, x_i, \lambda)).$$

Proof The statement follows immediately from the definition of the Kullback-Leibler divergence.

$$\begin{aligned}
\sum_{i=1}^{m} w_i \mathcal{D}_{KL}(q_i||p(\cdot|\, x_i, \lambda)) &= \sum_{i=1}^{m}\sum_{j=1}^{M} w_i q_{ij} \log \frac{q_{ij}}{p(c_j|\, x_i, \lambda)} \\
&= -\sum_{i=1}^{m}\sum_{j=1}^{M} w_i q_{ij} \log p(c_j|\, x_i, \lambda) + \sum_{i=1}^{m}\sum_{j=1}^{M} w_i q_{ij} \log q_{ij} \\
&= -l^U(\lambda : w, x, Y) - \sum_{i=1}^{m} w_i H(q_i),
\end{aligned}$$

where $H(q_i)$ is the entropy of the distribution q_i. The term $\sum_{i=1}^{m} H(q_i)$ does not depend on the model parameters and hence maximizing the the weighted uncertain log-likelihood is equivalent to minimizing the weighted average Kullback-Leibler divergence term. □

The decomposition (2.2) is extremely intuitive, with the uncertain likelihood maximization problem being now interpreted as a minimization of discrepancies between observed and model distributions. The uncertain maximum entropy model attempts to reproduce as faithfully as possible the randomness inherent in the observed sample, in contrast to standard (deterministic) likelihood maximization, which attempts to maximize conditional model probabilities corresponding to the classes y_i which were observed. Neal and Hinton (1998) give a similar result to (2.2), which hints at a link to a relaxed expectation-maximization type algorithm that considers the weights w_i to be expectations of unobserved random variables that should be updated incrementally at each iteration after the "maximization" step. We will later explore such an algorithm, although we will not stress this interpretation, instead showing convergence directly.

3. Noisy F-measure

It is already known (see Dimitroff et al., 2014) that maximization of expected F-measure can be performed via weighted maximum entropy in the case of deterministically labeled data. We extend this result to data with random labels, but first we must define one or more generalizations of the expected F-measure appropriate for this setting. We focus on binary classification, where we must classify examples into one of two classes, typically called the positive and negative class. In the deterministic case, we have the goodness-of-fit measures, precision P and recall R, with formulae given by

$$P = \frac{TP}{TP + FP}, \quad R = \frac{TP}{TP + FN}$$

(where TP, FN, FP and TN represent respectively the number of true positives, false negatives, false positives and true negatives returned by the model). The F-measure is the weighted harmonic mean of P and R, i.e.

$$F_\beta = \left(\frac{\beta}{P} + \frac{1-\beta}{R}\right)^{-1}.$$

In much of the literature, in particular van Rijsbergen (1979), the parameter β is instead called α. We go against this convention in keeping with the notation of Dimitroff et al. (2014).

In the presence of random labels, we have the underlying class distribution of training examples given by the matrix $Q = (q_{ij})$. When we maximize the uncertain likelihood function, we get model distributions $p(\cdot|\, x_i, \lambda)$, which we will sometimes express for convenience as $p_i(\lambda) = (p_{i1}(\lambda), \ldots, p_{iM}(\lambda))$, reducing to $(p_{i1}(\lambda), p_{i0}(\lambda))$ in the binary classification case, where as usual we give the positive class the label 1 and the negative class 0. In some situations to distinguish the class with label k from the integer k we will also use the label c_k.

In order to generalize the F-measure, we first define the "confusion" random variable $C_i = (D_i, M_i)$ where the marginal distributions of the first and second coordinates are equal to the data and the model distributions respectively. That is, D_i follows the Bernoulli distribution q_i and M_i represents the class returned by the model, i.e. is a random variable with distribution $p_i(\lambda)$. Noting that C_i can take four possible values, we are able to define equivalents to TP, FN, FP and TN as sums of the following quantities over i:

$$TP_i^U = P(C_i = (1,1)), \quad FN_i^U = P(C_i = (1,0)),$$
$$FP_i^U = P(C_i = (0,1)), \quad TN_i^U = P(C_i = (0,0)).$$

Clearly, to be able to calculate the above quantities we need to know the distribution of C_i, but we have only been provided with the marginals. To put it differently, the way that we calculate these quantities relies on what we consider to be the nature of the dependence between the true and modeled classes. The most straightforward approach would be to assume that the coordinates are independent, which yields products of the probabilities, for example $TP_i^U = P(C_i = (1,1)) = q_{i1}p_{i1}(\lambda) = q_{i1}p(c_1|\, x_i, \lambda)$. However, intuitively speaking, we do hope that the model prediction depends on the class distribution of each particular example. We can model (the strongest possible) dependence between the D_i's and the M_i's by setting

$$P(C_i = (1,1)) = \min(q_{i1}, p_{i1}(\lambda)).$$

Indeed, given the marginals $P(D_i = 1)$ and $P(M_i = 1)$, this corresponds to the strongest positive dependence, and in particular the strongest positive correlation, between the model and data, since

$P(M_i = 1 \,|\, D_i = 1) = P(C_i = (1,1))/P(D_i = 1)$, and obviously $P(M_i = 1 \,|\, D_i = 1)$ is largest when $P(C_i = (1,1))$ is largest. On the other hand, $P(C_i = (1,1)) \leq \min(P(D_i = 1), P(M_i = 1))$. Using the above chosen value for $P(C_i = (1,1))$ it is straightforward to obtain the rest of the probability weights describing the distribution of C_i. This leaves us with:

$$\begin{aligned} TP_i^U &= P(C_i = (1,1)) = \min(q_{i1}, p_{i1}(\lambda)) \\ FN_i^U &= P(C_i = (1,0)) = \max(0, q_{i1} - p_{i1}(\lambda)) = \max(0, p_{i0}(\lambda) - q_{i0}) \\ FP_i^U &= P(C_i = (0,1)) = \max(0, p_{i1}(\lambda) - q_{i1}) \\ TN_i^U &= P(C_i = (0,0)) = \min(q_{i0}, p_{i0}(\lambda)) \end{aligned} \tag{3.1}$$

Note that taking the minima of probabilities in the true classification case allows us to take only the probability weight common to the true and model distributions, whereas we only get weight for false classification if we predict there to be more, or less, probability weight in the given class than actually exists. This property is highly desirable and natural.

The above definitions (3.1), are confusion contributions for the i'th training example. The uncertain equivalents to the deterministic TP, FN, FP and TN confusion counts are the sums of our newly defined quantities over i. In the case of true and modeled classes being considered dependent, we get:

$$\begin{aligned} TP^U &= \sum_{i=1}^{m} \min(q_{i1}, p(c_1 |\, x_i, \lambda)) \\ FN^U &= \sum_{i=1}^{m} \max(0, p(c_0 |\, x_i, \lambda) - q_{i0}) \\ FP^U &= \sum_{i=1}^{m} \max(0, p(c_1 |\, x_i, \lambda) - q_{i1}) \\ TN^U &= \sum_{i=1}^{m} \min(q_{i0}, p(c_0 |\, x_i, \lambda)) \end{aligned}$$

Using these definitions, we derive noisy Precision and noisy Recall:

$$P^U = \frac{TP^U}{TP^U + FP^U}, \quad R^U = \frac{TP^U}{TP^U + FN^U}$$

and the noisy F–measure follows:

$$F_\beta^U = \left(\frac{\beta}{P^U} + \frac{1-\beta}{R^U} \right)^{-1} = \frac{TP^U}{\beta(TP^U - TN^U) + \beta \sum_{i=1}^m q_{i0} + (1-\beta) \sum_{i=1}^m q_{i1}}.$$

In the uncorrelated case the definitions are analogous, but with different distributions for the C_i's. In either the correlated or uncorrelated case, we wish to directly maximize the noisy F-measure. It can already be seen that a disadvantage of the correlated F-measure is that it is not differentiable everywhere, however the singularity is rather moderate as the derivatives exist in the weak distributional sense and are proper functions. Therefore, all the calculations do work out with only a minor additional complexity. However, the advantage, due to its definitional properties, is

that maximization of correlated F-measure corresponds roughly also to a minimization of L^1 discrepancies between q_i and $p_i(\lambda)$. Indeed, maximization of uncorrelated F-measure has no such interpretation, and corresponds instead to maximizing/minimizing model probabilities as in the deterministic case, instead of attempting to emulate the true distribution. This is a major deficiency of the independent generalization of the F-measure (provided one expects the model to follow the distribution of the data and instead of trying to reproduce the class with the largest probability) and therefore we do not pursue it in what follows but rather we focus on the correlated F-measure.

One might rightfully argue that the above discussed two alternatives represent rather extreme assumptions – either independence or the strongest possible dependence between true and model classes. In a straightforward manner we may define all other cases as follows. For a parameter $\theta \in [0, 1]$, we set:

$$P(C_i = (1, 1)) = q_{i1}p_{i1}(\lambda) + \theta(\min(q_{i1}, p_{i1}(\lambda)) - q_{i1}p_{i1}(\lambda)).$$

All of the other probabilities $P(C_i = (0, 1))$, $P(C_i = (1, 0))$ and $P(C_i = (0, 0))$ follow immediately, and for different θ we obtain a different level of dependence. Obviously with $\theta = 0$ we get the independent case and with $\theta = 1$ the strongest possible dependence. For the sake of simplicity of the exposition, we will not follow this general approach as it is at least on a technical level the same as the extreme case with $\theta = 1$. Still, we want to stress that the approach we follow would yield an algorithm for the maximization of this generalization of the F-measure as well.

4. Maximizing the noisy F-measure via weighted uncertain likelihood

We present now our main result, which directly links noisy F-measure maximization and uncertain likelihood maximization.

Proposition 4.1. *Let $\hat{\lambda}_\beta$ be the maximizer of the noisy F-measure F_β^U. Then there exists a vector of weights $w(\beta) \in \mathbb{R}^m$ such that $\hat{\lambda}_\beta$ coincides with the weighted maximum uncertain likelihood estimator*

$$\hat{\lambda}_{ML}^{w(\beta)} = \arg\max_\lambda l^U(\lambda : w(\beta), x, Y).$$

That is, we have

$$\hat{\lambda}_\beta = \hat{\lambda}_{ML}^{w(\beta)}.$$

Proof Let $\hat{\lambda}_\beta$ be the F_β^U maximizer, i.e.

$$\hat{\lambda}_\beta = \arg\max_\lambda F_\beta^U.$$

When we give the following form of the noisy F-measure:

$$F_\beta^U = \frac{TP^U}{\beta(TP^U - TN^U) + \beta\sum_{i=1}^m q_{i0} + (1-\beta)\sum_{i=1}^m q_{i1}},$$

we realize that the maximizer $\hat{\lambda}_\beta$ of F_β^U is an element of the Pareto optimal set of the multi-criteria optimization problem (MOP):

$$\max_\lambda \left\{TP^U, TN^U\right\},$$

or:

$$\max_{\lambda} \left\{ \sum_{i=1}^{m} \min(q_{i1}, p(c_1 | x_i, \lambda)), \sum_{i=1}^{m} \min(q_{i0}, p(c_0 | x_i, \lambda)) \right\}. \tag{4.1}$$

This is because the denominator of F_β^U is always positive and the function

$$f(x, y) = \frac{x}{\beta(x - y) + \beta \sum_{i=1}^{m} q_{i0} + (1 - \beta) \sum_{i=1}^{m} q_{i1}}$$

is increasing in x and y. If we assume that $\hat{\lambda}_\beta$ is not Pareto efficient for the MOP (4.1), then we can find another set of parameters λ_0 such that the pair $(TP^U(\lambda_0), TN^U(\lambda_0))$ dominates $(TP^U(\hat{\lambda}_\beta), TN^U(\hat{\lambda}_\beta))$, that is, at least one of the objectives is improved in λ_0 when compared with $\hat{\lambda}_\beta$ with the other one having not decreased. However, this would mean that $F_\beta^U(\lambda_0) > F_\beta^U(\hat{\lambda}_\beta)$, which contradicts the assumption that $\hat{\lambda}_\beta$ maximizes the noisy F-measure.

Passing to the finer granularity MOP, we observe also that $\hat{\lambda}_\beta$ must be an element of the Pareto optimal set of:

$$\max_{\lambda} \{\min(q_{i1}, p(c_1 | x_i, \lambda)), \min(q_{i0}, p(c_0 | x_i, \lambda)) : \forall i\}. \tag{4.2}$$

Indeed, the following argument shows that (4.2) has a Pareto optimal set which contains that of (4.1). If we assume that λ is Pareto optimal for (4.1) but not for (4.2), then we can find λ_0 such that some of the objectives (4.2) are improved and none of them is decreased. But this would mean that the pair $(TP^U(\lambda_0), TN^U(\lambda_0))$ dominates $(TP^U(\lambda), TN^U(\lambda)$, which directly contradicts the assumption of Pareto optimality of λ for (4.1). Therefore Pareto optimality for (4.1) implies Pareto optimality for (4.2).

See that the Pareto optimal set of (4.2) is the same as the Pareto optimal set of:

$$\begin{aligned} \max_{\lambda} \{ & [q_{i1} \log p(c_1 | x_i, \lambda) + q_{i0} \log p(c_0 | x_i, \lambda)] \, \mathbb{1}\{p(c_1 | x_i, \lambda) \leq q_{i1}\}, \\ & [q_{i1} \log p(c_1 | x_i, \lambda) + q_{i0} \log p(c_0 | x_i, \lambda)] \, \mathbb{1}\{p(c_1 | x_i, \lambda) \geq q_{i1}\} : \forall i \} \end{aligned} \tag{4.3}$$

which is equivalent to:

$$\begin{aligned} \max_{\lambda} \{ & - [H(q_i) + \mathcal{D}_{KL}(q_i || p(\cdot | x_i, \lambda))] \, \mathbb{1}\{p(c_1 | x_i, \lambda) \leq q_{i1}\}, \\ & - [H(q_i) + \mathcal{D}_{KL}(q_i || p(\cdot | x_i, \lambda))] \, \mathbb{1}\{p(c_1 | x_i, \lambda) \geq q_{i1}\} : \forall i \}. \end{aligned} \tag{4.4}$$

We are able to claim that the Pareto optimal set of (4.2) is the same as that of (4.4) by the following argument. Take λ which is Pareto optimal for (4.2) but not for (4.4). Then in (4.4) there is a λ_0 such that for at least one i, we have one of the given objectives improving on λ without the other one decreasing. By the form of the objectives (negative Kullback-Leibler divergences), this means that $|q_{i1} - p(c_1 | x_i, \lambda_0)| < |q_{i1} - p(c_1 | x_i, \lambda)|$. But then in (4.2) we have by the form of the $\min$ function that λ_0 improves upon the i'th objectives in λ in the same way, i.e. λ is not Pareto optimal for (4.2). We conclude by this contradiction that $\hat{\lambda}_\beta$ is in the Pareto optimal set of (4.4). Hence we have shown that the Pareto optimal set of (4.2) is contained in the Pareto optimal set of (4.4). Showing the opposite inclusion is analogous, thus (4.2) and (4.4) have the same Pareto optimal set.

Since the objectives in (4.4) are concave, every point in its Pareto optimal set can be represented as a maximizer of:

$$G(\lambda : u, v, x, Y) = -\sum_{i=1}^{m} \{u_i \left[H(q_i) + \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i, \lambda))\right] \mathbb{1}\{p(c_1|\,x_i, \lambda) \leq q_{i1}\} + v_i \left[H(q_i) + \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i, \lambda))\right] \mathbb{1}\{p(c_1|\,x_i, \lambda) \geq q_{i1}\}\} \quad (4.5)$$

for some nonnegative weights $u, v \in \mathbb{R}^m$. This follows from the result given for example by Ehrgott (2005) that all solutions of a MOP can be obtained by maximizing nonnegative linear combinations of the objectives.

So $\hat{\lambda}_\beta$ can be represented as a maximizer of G for some u, v. G can be viewed in the following way:

$$G(\lambda : w, x, Y) = -\sum_{i=1}^{m} w_i \left[H(q_i) + \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i, \lambda))\right],$$

where

$$w_i = u_i \mathbb{1}\{p(c_1|\,x_i, \lambda) \leq q_{i1}\} + v_i \mathbb{1}\{p(c_1|\,x_i, \lambda) \geq q_{i1}\}.$$

Note that these weights w_i still depend on the unknown λ. We calculate them as in the deterministic case by matching the coefficients of gradients.

We know that $\nabla F_\beta^U(\hat{\lambda}_\beta) = 0$. Now:

$$\nabla F_\beta^U = \sum_{i=1}^{m} \frac{\partial F_\beta^U}{\partial TP^U} \cdot \frac{\partial TP^U}{\partial p(c_1|\,x_i, \lambda)} \nabla p(c_1|\,x_i, \lambda) + \sum_{i=1}^{m} \frac{\partial F_\beta^U}{\partial TN^U} \cdot \frac{\partial TN^U}{\partial p(c_1|\,x_i, \lambda)} \nabla p(c_1|\,x_i, \lambda).$$

The derivatives of TP^U and TN^U are generalized derivatives, since TP^U and TN^U are continuous when taken as functions of $p(c_1|\,x_i, \lambda)$ which are not smooth only at one point ($p(c_1|\,x_i, \lambda) = q_{i1}$). We get:

$$\frac{\partial TP^U}{\partial p(c_1|\,x_i, \lambda)} = \frac{\partial \min(q_{i1}, p(c_1|\,x_i, \lambda))}{\partial p(c_1|\,x_i, \lambda)} = \mathbb{1}\{p(c_1|\,x_i, \lambda) \leq q_{i1}\}$$
$$\frac{\partial TN^U}{\partial p(c_1|\,x_i, \lambda)} = \frac{\partial \min(1 - q_{i1}, 1 - p(c_1|\,x_i, \lambda))}{\partial p(c_1|\,x_i, \lambda)} = -\mathbb{1}\{p(c_1|\,x_i, \lambda) \geq q_{i1}\}.$$

and

$$\nabla F_\beta^U = \sum_{i=1}^{m} \left[\frac{\partial F_\beta^U}{\partial TP^U} \mathbb{1}\{p(c_1|\,x_i, \lambda) \leq q_{i1}\} - \frac{\partial F_\beta^U}{\partial TN^U} \mathbb{1}\{p(c_1|\,x_i, \lambda) \geq q_{i1}\}\right] \nabla p(c_1|\,x_i, \lambda). \quad (4.6)$$

We also have the alternate representation of $\hat{\lambda}_\beta$ as a maximizer of G. This means that $\nabla G(\hat{\lambda}_\beta) = 0$ for some correct choice of $u(\beta), v(\beta)$ to be determined.

$$\nabla G = -\sum_{i=1}^{m} \left\{u(\beta)_i \frac{\partial \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i, \lambda))}{\partial p(c_1|\,x_i, \lambda)} \mathbb{1}\{p(c_1|\,x_i, \lambda) \leq q_{i1}\} + v(\beta)_i \frac{\partial \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i, \lambda))}{\partial p(c_1|\,x_i, \lambda)} \mathbb{1}\{p(c_1|\,x_i, \lambda) \geq q_{i1}\}\right\} \nabla p(c_1|\,x_i, \lambda) \quad (4.7)$$

Comparing gradients, we find appropriate weights, to be evaluated at $\lambda = \hat{\lambda}_\beta$:

$$u(\beta)_i = -\frac{\partial F_\beta^U}{\partial TP^U} \Bigg/ \frac{\partial \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i,\lambda))}{\partial p(c_1\,|\,x_i,\lambda)}$$
$$v(\beta)_i = \frac{\partial F_\beta^U}{\partial TN^U} \Bigg/ \frac{\partial \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i,\lambda))}{\partial p(c_1\,|\,x_i,\lambda)}.$$

Observe that by letting

$$w(\beta)_i = u(\beta)_i \mathbb{1}\{p(c_1\,|\,x_i,\hat{\lambda}_\beta) \leq q_{i1}\} + v(\beta)_i \mathbb{1}\{p(c_1\,|\,x_i,\hat{\lambda}_\beta) \geq q_{i1}\}$$

we arrive at the uncertain likelihood

$$G(\lambda : w(\beta), x, Y) = l^U(\lambda : w(\beta), x, Y) = -\sum_{i=1}^{m} w(\beta)_i \left[H(q_i) + \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i,\lambda))\right].$$

Note that this choice of $w(\beta)$ is always positive.

After some simple algebra, we find that:

$$\frac{\partial F_\beta^U}{\partial TP^U} = \frac{F_\beta^U}{TP^U}\left[1 - \beta F_\beta^U\right]$$
$$\frac{\partial F_\beta^U}{\partial TN^U} = \frac{\beta (F_\beta^U)^2}{TP^U},$$

and

$$\frac{\partial \mathcal{D}_{KL}(q_i||p(\cdot\,|\,x_i,\lambda))}{\partial p(c_1\,|\,x_i,\lambda)} = \frac{p(c_1\,|\,x_i,\lambda) - q_{i1}}{p(c_1\,|\,x_i,\lambda)\cdot p(c_0\,|\,x_i,\lambda)}.$$

Our final expression for the weights $w(\beta)$ is:

$$w(\beta)_i = \frac{p(c_1\,|\,x_i,\hat{\lambda}_\beta)\cdot p(c_0\,|\,x_i,\hat{\lambda}_\beta)}{p(c_1\,|\,x_i,\hat{\lambda}_\beta) - q_{i1}} \cdot \frac{F_\beta^U(\hat{\lambda}_\beta)}{TP^U(\hat{\lambda}_\beta)} \Big[\beta F_\beta^U(\hat{\lambda}_\beta)\mathbb{1}\{p(c_1\,|\,x_i,\hat{\lambda}_\beta) \geq q_{i1}\}$$
$$-(1-\beta F_\beta^U(\hat{\lambda}_\beta))\mathbb{1}\{p(c_1\,|\,x_i,\hat{\lambda}_\beta) \leq q_{i1}\}\Big] \quad (4.8)$$

Each F_β^U maximizer can be realized as a maximizer of the weighted uncertain likelihood, for these weights $w(\beta)$. □

5. Algorithm

The important takeaway from the previous section is that each maximizer of the expected correlated F_β measure can also be realized as a weighted uncertain likelihood maximizer. Since the uncertain likelihood inherits nice properties from the standard likelihood, and in particular concavity, it is simple to maximize. When we turn our attention to the form of the weights (4.8), we realize that as in the deterministic likelihood case, they depend on the parameter $\hat{\lambda}_\beta$ we wish to obtain, which is at first sight unfortunate. However, we may determine them adaptively via the following algorithm. Note that in (4.8), the denominator includes a factor of $p(c_1\,|\,x_i,\hat{\lambda}_\beta) - q_{i1}$ and so, when theoretical and model distributions are close (which ideally they are), the weights may explode in size. To control this behaviour in practice we place computationally sensible bounds on $w(\beta)$.

Algorithm 1 F_β^U maximizer

1: $n = 1$.
2: Initialize model parameters and calculate initial $\hat{F}_\beta^U(1)$ and $\hat{TP^U}(1)$ from initialized model.
3: Set weights (with some imposed bound)

$$(w_n)_i = \frac{p(c_1|\, x_i, \hat{\lambda}_n) \cdot p(c_0|\, x_i, \hat{\lambda}_n)}{p(c_1|\, x_i, \hat{\lambda}_n) - q_{i1}} \cdot \frac{\hat{F}_\beta(n)}{\hat{TP^U}(n)} \Big[\beta \hat{F}_\beta(n) \mathbb{1}\{p(c_1|\, x_i, \hat{\lambda}_n) \geq q_{i1}\} - (1 - \beta \hat{F}_\beta(n)) \mathbb{1}\{p(c_1|\, x_i, \hat{\lambda}_n) \leq q_{i1}\}\Big].$$

4: Do one full-batch update of weighted uncertain log-likelihood maximization with weights w_n. $n := n + 1$.
5: Calculate model $\hat{F}_\beta^U(n)$, $\hat{TP^U}(n)$ and model conditional probabilities $p(y_i \,|\, x_i, \hat{\lambda}_n)$.
6: If convergence criteria not met (no significant improvement of target F_β^U in the last k steps) $\rightarrow$ Step 3.

5.1 Convergence of the algorithm

One of the strengths of the algorithm is that it is straightforward to implement. The weighted uncertain log-likelihood maximization proceeds similarly to weighted log-likelihood maximization, which is analogous to the standard one with a different gradient involving the weights. The implementation then follows some off-the-shelf version of the gradient ascent algorithm.

We now show that, given that the learning rate for the full-batch gradient ascent is small enough, each step of Algorithm 1 improves the attained value of the noisy F-measure. Hence, if the learning rate is decreasing appropriately, the algorithm will converge.

At the n'th step of the algorithm, the following update of the parameters λ is performed:

$$\hat{\lambda}_{n+1} := \hat{\lambda}_{n+1} + \epsilon_n \cdot \nabla l^U(\lambda_n : w_n, x, y) \tag{5.1}$$

where w_n are the weights described in the algorithm, and ϵ_n is the learning rate. The update (5.1) describes a theoretically small change in λ_{n+1} in the direction of the gradient at the previous λ_n.

For the gradient ∇l^U, from (2.1) we have:

$$\nabla l^U(\lambda : w_n, x, Y) = \sum_{i=1}^{m} (w_n)_i \frac{q_{i1} - p(c_1|\, x_i, \lambda)}{p(c_1|\, x_i, \lambda) \cdot p(c_0|\, x_i, \lambda)} \nabla p(c_1|\, x_i, \lambda)$$

Then using the definition of w_n,

$$\nabla l^U(\lambda : w_n, x, Y) = \sum_{i=1}^{m} \frac{p(c_1|\, x_i, \hat{\lambda}_n) \cdot p(c_0|\, x_i, \hat{\lambda}_n)}{p(c_1|\, x_i, \lambda) \cdot p(c_0|\, x_i, \lambda)} \cdot \frac{p(c_1|\, x_i, \lambda) - q_{i1}}{p(c_1|\, x_i, \hat{\lambda}_n) - q_{i1}} \times \left[\frac{\partial F_\beta^U}{\partial TP^U} \mathbb{1}\{p(c_1|\, x_i, \lambda) \leq q_{i1}\} - \frac{\partial F_\beta^U}{\partial TN^U} \mathbb{1}\{p(c_1|\, x_i, \lambda) \geq q_{i1}\}\right] \nabla p(c_1|\, x_i, \lambda).$$

Now from (4.6), evaluating the above gradient at the current parameter state $\hat{\lambda}_n$ obviously yields:

$$\nabla l^U(\hat{\lambda}_n : w_n, x, Y) = \nabla F_\beta^U(\hat{\lambda}_n).$$

Therefore, the update (5.1) always results in an improvement of the objective F_β^U. This proves that the algorithm will eventually converge to a local maximum of noisy F-measure. The immediate question arises of finding the global maximum of noisy F-measure, and whether at least a better local maximum can be found using Algorithm 1 by performing at each iteration more than one full-batch update of uncertain likelihood maximization (i.e. more than one step of gradient ascent). We provide further motivation for exploring this direction of research in the next section, where it is noted that both convergence and performance tend to improve when considering more than one step of gradient ascent. It might be that this revised approach allows us to escape regions with suboptimal local maxima of F_β^U. If this is the case, the question remains of how to choose the number of steps in the gradient ascent performed at each iteration of our algorithm.

6. Experiments

We now test experimentally the performance of noisy F-measure optimization via our algorithm. Our aim is to show that uncertain likelihood maximization works successfully, and that we can improve its performance in terms of noisy F-measure via Algorithm 1. We test our algorithm on four different datasets, which are as follows.

Synthetic data – A We simulate a dataset of 1100 samples (x_i, y_i) with two classes, u (positive class) and $\bar{u}$ (negative class), and two features. Each class contains 550 samples, distributed as spherical Gaussians in the space of features. The samples from class $\bar{u}$ are distributed as $\mathcal{N}(\mu_0, \Sigma_0)$, where $\mu_0 = (0.5, 1)$ and $\Sigma_0 = (0.3, 0.3)^T I_2$. Class u is generated by $\mathcal{N}(\mu_1, \Sigma_1)$, with $\mu_1 = (1, 0.8)$ and $\Sigma_1 = (0.3, 0.3)^T I_2$. We use 1000 examples for training and 100 for testing.

Synthetic data – B We simulate a dataset of 1100 samples (x_i, y_i) with two classes, u and $\bar{u}$, and two features. Each class contains 550 samples, distributed as elliptical Gaussians in the space of features. The samples from class $\bar{u}$ are distributed as $\mathcal{N}(\mu_0, \Sigma_0)$, where $\mu_0 = (2, 1)$ and $\Sigma_0 = (1, 0.3)^T I_2$. Class u is generated by $\mathcal{N}(\mu_1, \Sigma_1)$, with $\mu_1 = (1, 2)$ and $\Sigma_1 = (0.3, 1)^T I_2$. We use 1000 examples for training and 100 for testing.

Titanic data We consider the famous "Titanic" dataset (taken from Vanderbilt University (2014)) with 1309 samples, two classes and five features. There are 500 samples in the positive class and 809 samples in the negative class. We use 1000 examples for training and 309 for testing. The data itself concerns the survival of passengers on the Titanic. The positive class contains passengers who survived and the negative class contains passengers who did not survive. The features we use are passenger class, age, sex and number of siblings/spouses aboard, as well as an interaction effect between passenger class and sex, since it is well-known that in the sinking of the Titanic, passenger class showed a larger difference in survival rate amongst women compared to men (that is, women of a higher passenger class were much more likely to survive than those of a lower class, whereas this division was more marginal amongst men). It should be noted that we present this dataset only as an example of data with some non-trivial features to demonstrate the "natural" performance of our algorithm. That is, we could give the uncertain interpretation of the Titanic dataset to represent a short-term after the fact measure of certainty of whether a given passenger is currently alive or dead, but we mostly eschew such interpretation, focusing instead directly on performance, which itself gives an indication of the potential of our method to be used for much more practical applications in the future.

SPECT data We consider the SPECT heart dataset (provided by Bache and Lichman, 2013) with 267 samples, two classes and 22 features. There are 212 samples in the positive class and 55 samples in the negative class. We use 235 examples for training and 32 for testing. The data represents Single Proton Emission Computed Tomography (SPECT) images of human hearts, from which 22 binary features have been extracted. Each patient is also classified into one of two categories, normal and abnormal (positive and negative class respectively). The uncertain interpretation of this dataset is that the diagnosis of a patient is in many cases unable to be known definitively, even if it is available with a high probability. When diagnosis is to be performed on the basis of SPECT images, such information may easily be open to multiple interpretations.

We now elaborate on the precise nature of our experiments. We take a set of data with theoretically deterministic classes as above. We then define new positive and negative classes c_1 and c_0, and the class random variables $Y_i \sim q_i$ corresponding to each training example and for some given matrix of class Bernoulli distributions q. Actually, we generate this prior probability matrix q ourselves in the following way. We consider class-flipping probabilities (positive u to negative c_0 or negative $\bar{u}$ to positive c_1) given by arbitrary β distributions. That is, we flip points x_i which theoretically belong to class u into class c_0 with probabilities $P(Y_i = c_0 | \, y_i = u) = q_{i0} \sim \beta(1, 5)$, and we flip points x_i which theoretically belong to class $\bar{u}$ into class c_1 with probabilities $P(Y_i = c_1 | \, y_i = \bar{u}) = q_{i1} \sim \beta(2, 5)$. Now we have the uncertain training set (x_i, Y_i) with given prior class probability matrix q.

We visualize the most likely deterministic realization of the above on samples from datasets A and B in Figure 1, using 1000 data points. That is, for the purposes of visualization, we fix a deterministic class for each example x_i corresponding to the higher value of $q_i = (q_{i1}, q_{i0})$

Of course, other choices of class-wise noise are possible. We also give results for one-sided noise on the Titanic dataset, illustrating the success of our algorithm in this setting. In this case, we are only generating the negative-to-positive class-flipping noise mentioned above, and ignore the other direction.

As discussed earlier, the main interpretation we give to class uncertainty is that we have a panel of annotators, some of whom thought that the point x_i belonged to the class c_1 and others who thought that x_i belonged to c_0. The proportion of opinions is then given by row q_i. Within this realistic framework, we consider there to be no necessary "ground truth" (even though originally we contrived the data as having belonged to certain true distributions).

In the absence of the results of uncertain classification presented in this thesis, we would ordinarily perform deterministic maximum entropy classification, setting for example:

$$y_i = \begin{cases} 1 & \text{if } q_{i1} \geq 0.5 \\ 0 & \text{if } q_{i0} > 0.5. \end{cases} \tag{6.1}$$

(We could alternatively apply some threshold other than 0.5.) We might even think naturally to fit a weighted maximum entropy based on the certainty of our observations, i.e. assigning weights $w_i = \max(q_{i1}, q_{i0})$, or some similar setup that gives more weight to data points with lower class entropy and less weight to data points with higher class entropy. There are a few possible ways of going about this, which we need not really consider, since our interest is in maximizing the noisy F-measure and we have an algorithm to do it. The details are as follows, and we perform each of the below for a range of $\beta \in [0, 1]$.

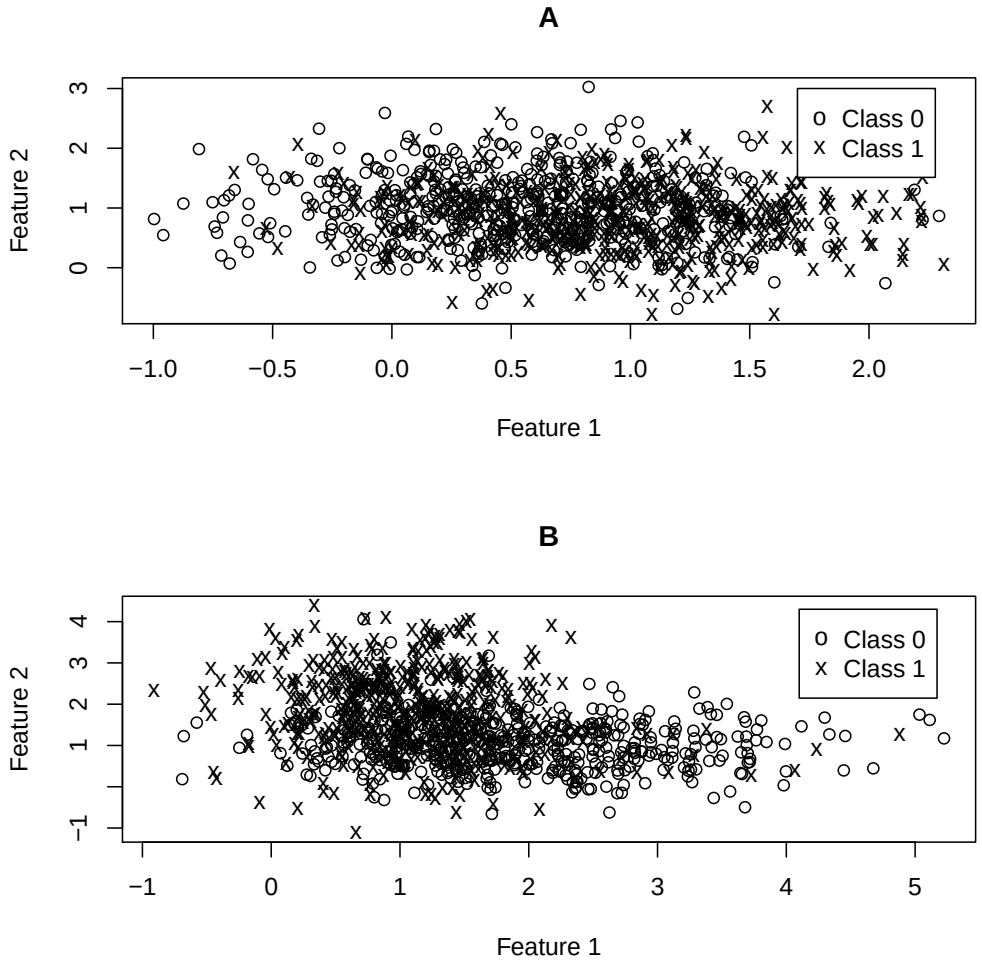

Figure 1: Most likely realization of samples in the space of features for synthetic datasets A and B.

We fit four models – deterministic maximum entropy (via thresholding as in (6.1)), F-measure maximizing deterministic maximum entropy (again via thresholding), uncertain maximum entropy and noisy F-measure maximizing uncertain maximum entropy. We consider the performance of each of these models on the training and test set, with respect to both F-measure and noisy F-measure. We expect that the deterministic models will perform well with respect to deterministic F-measure and the uncertain models will perform well with respect to noisy F-measure, since the deterministic models attempt to maximize model probabilities corresponding to deterministic classes, and the uncertain models attempt to minimize the discrepancy between model class distributions and given experimental class distributions.

Additionally, we sample $B = 1000$ times from the class distributions q and test the average deterministic F-measure of the fitted classifiers over all resamples. The intuition here is that an uncertain classifier, trained with the knowledge of the randomness inherent in the training sample, will give a generally better long-term deterministic F-measure on test data drawn from these distributions. If we return again to the interpretation of the uncertain likelihood as a deterministic likelihood of an infinite number of samples from the Bernoulli distributions q_i of random classes Y_i, the F-measure of this deterministic likelihood maximizer should approximate the noisy F-measure of the uncertain likelihood maximizer. This means indeed that noisy F-measure maximization should give optimal deterministic F-measure on a large number of samples from the distributions q.

We now provide the results of the experiments described. In Figure 2, we compare the performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the performance of the F-measure maximizing deterministic classifier and the noisy F-measure maximizing uncertain classifier on data A. The noisy F-measure maximizing uncertain maximum entropy is giving a significantly higher noisy F-measure on the training set than any of the other classifiers. While its performance in terms of F-measure is comparable with that of the deterministic F-measure maximizer for some values of β, it is clearly not performing to the same standard here, as expected.

In Figure 3, we compare the predictive, i.e. test set, performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the predictive performance of the F-measure maximizing deterministic classifier and the noisy F-measure maximizing uncertain classifier on data A. We see that indeed the F_β^U maximizer is giving improved performance on the test set over the baseline, and is able to reproduce the class randomness inherent in the training distributions in a very useful manner for predictive purposes. We observe that both on the training and test set, there is a point around $\beta = 0.6$ where the performance of our algorithm approximately meets that of its baseline. Additionally, while we did not expect overfitting to occur, it may be possible to be further minimized by introducing a regularization term into the uncertain likelihood. We leave this extension for the future.

In Figure 4, we test the average performance of our classifiers on $B = 1000$ datasets resampled via the distributions q. We have here fitted the uncertain likelihood classifier with weights calculated by plugging deterministic confusion counts and F-measure in the place of their uncertain counterparts. The improvement in using the noisy F-measure maximizing algorithm instead of the standard uncertain likelihood is seen particularly emphatically when we do this, evident for most values of β. The intuition behind the plug-in algorithm is simply that we are tailoring it towards the deterministic measure while approximately reproducing the observed class randomness.

In Figure 5, we compare the performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the performance of the F-measure maximizing determinis-

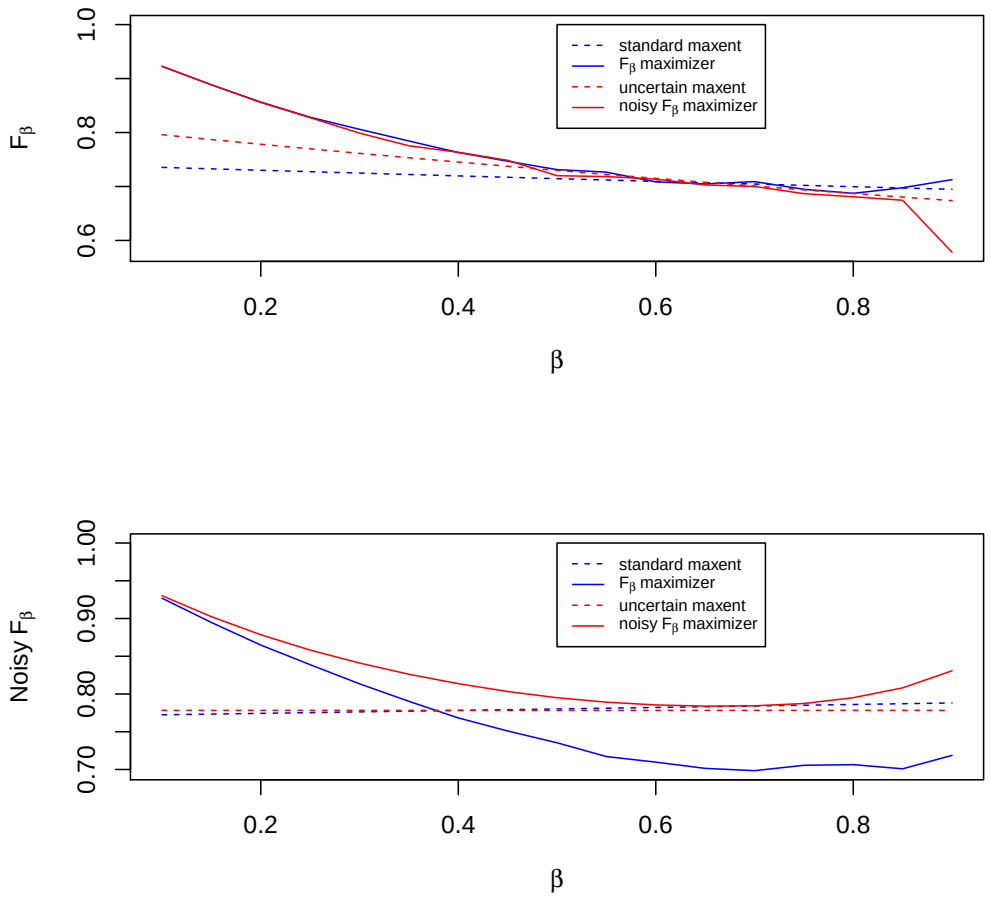

Figure 2: F_β and F_β^U on the training set for synthetic data A.

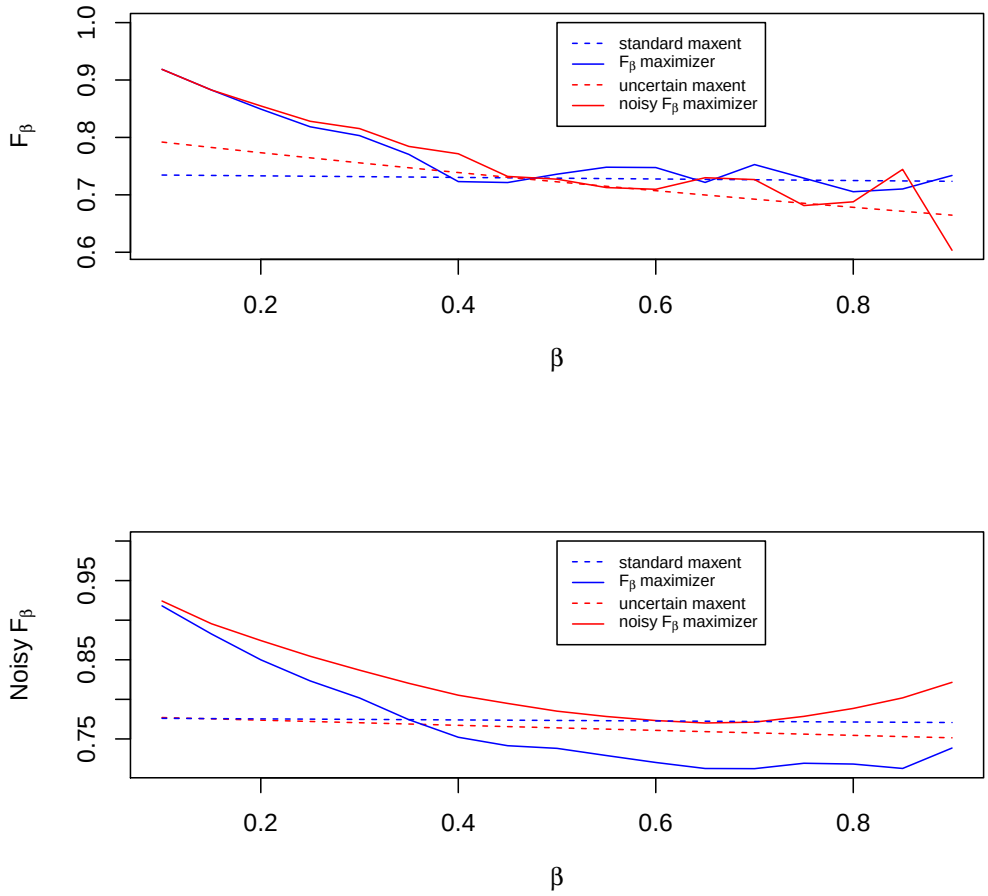

Figure 3: F_β and F_β^U on the test set for synthetic data A.

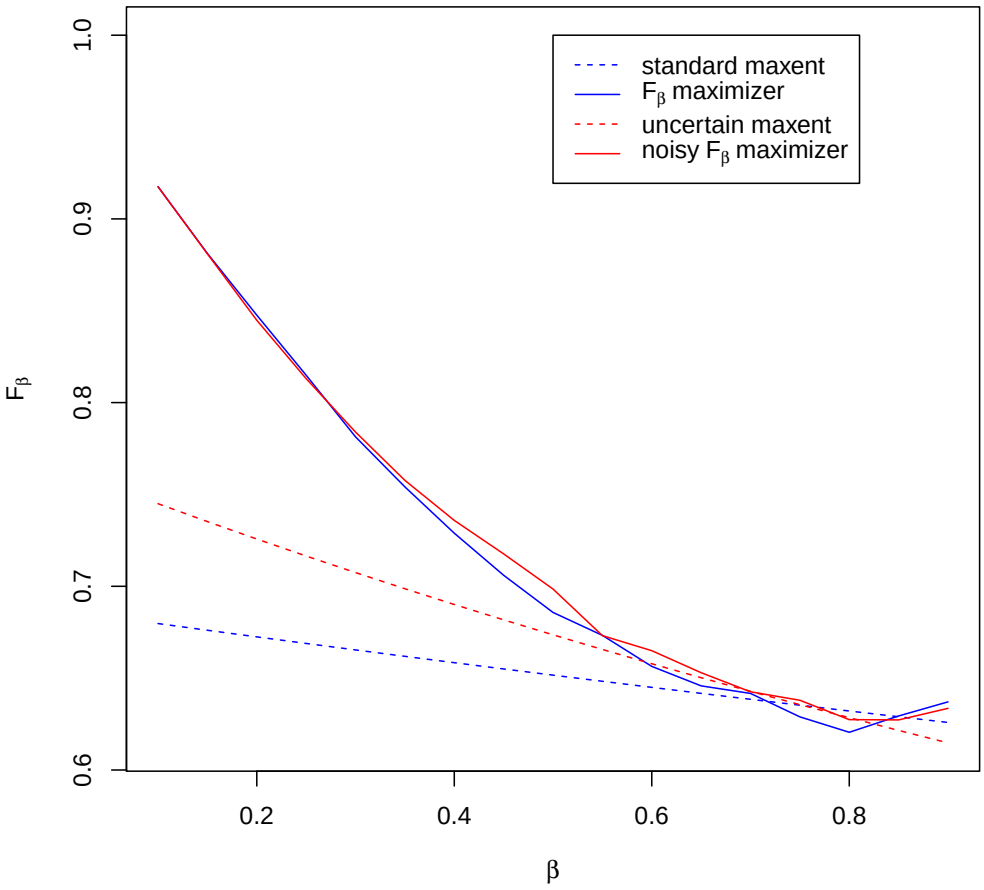

Figure 4: Average F_β on 1000 resamples from q on synthetic data A, with F_β^U maximizer fitted with weights chosen via deterministic F_β plug-in algorithm.

tic classifier and the noisy F-measure maximizing uncertain classifier on data B. In Figure 6, we compare the predictive, i.e. test set, performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the predictive performance of the F-measure maximizing deterministic classifier and the noisy F-measure maximizing uncertain classifier on data B. In Figure 7, we test the average performance of our classifiers on $B = 1000$ datasets resampled via the distributions q. We have again fitted the uncertain likelihood classifier with weights calculated by plugging deterministic confusion counts and F-measure in the place of their uncertain counterparts. The improvement in using the noisy F-measure maximizing algorithm instead of the standard uncertain likelihood is seen particularly emphatically when we do this, evident for most values of β.

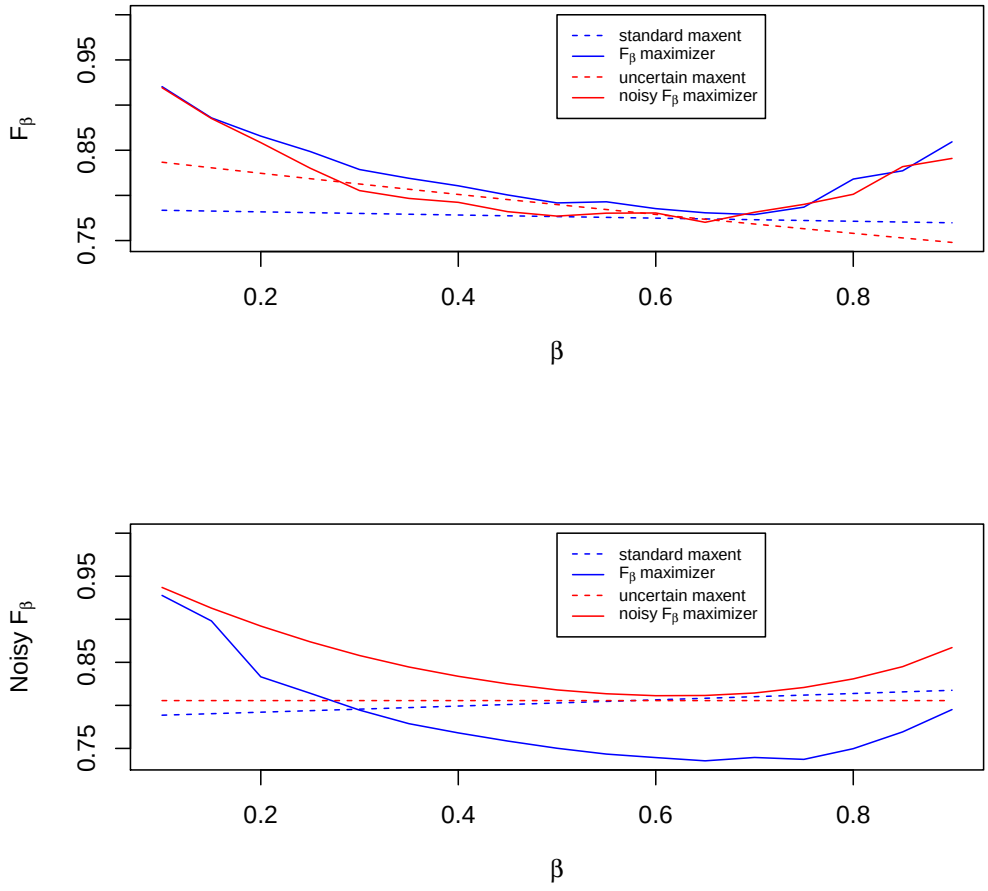

Figure 5: F_β and F_β^U on the training set for synthetic data B.

The conclusions for these results in the case of data B are basically identical to those we made for data A.

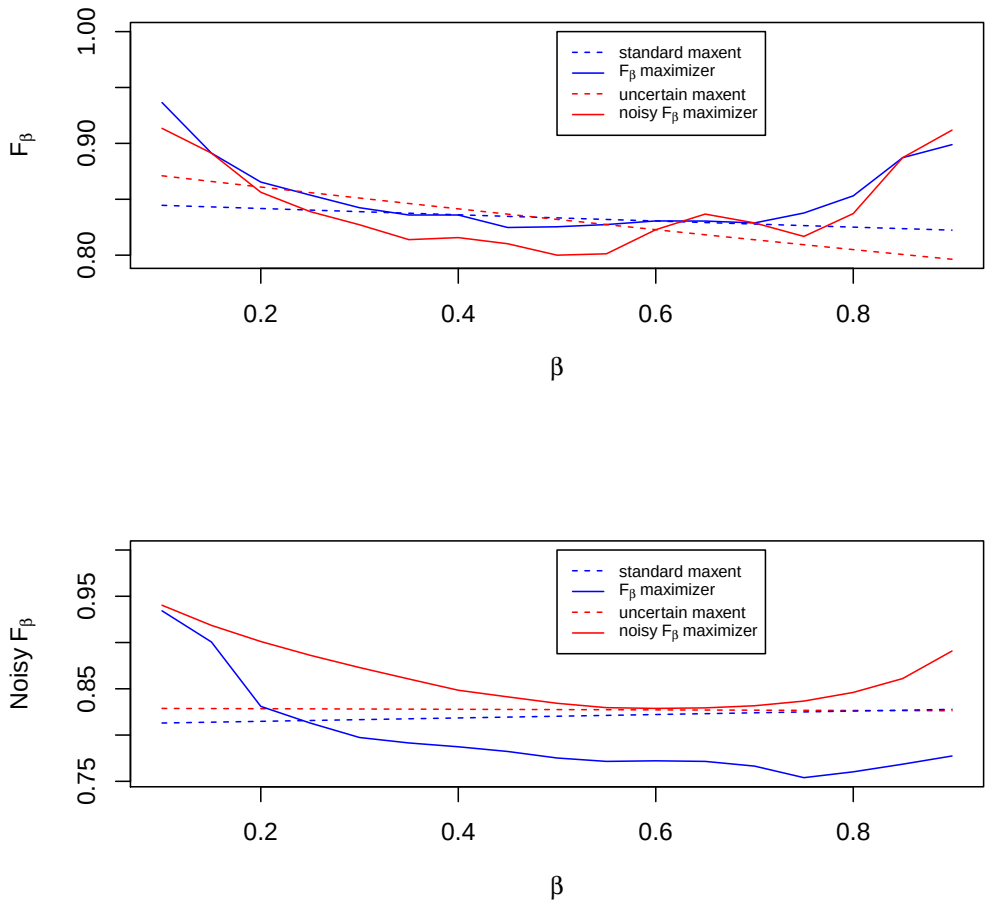

Figure 6: F_β and F_β^U on the test set for synthetic data B.

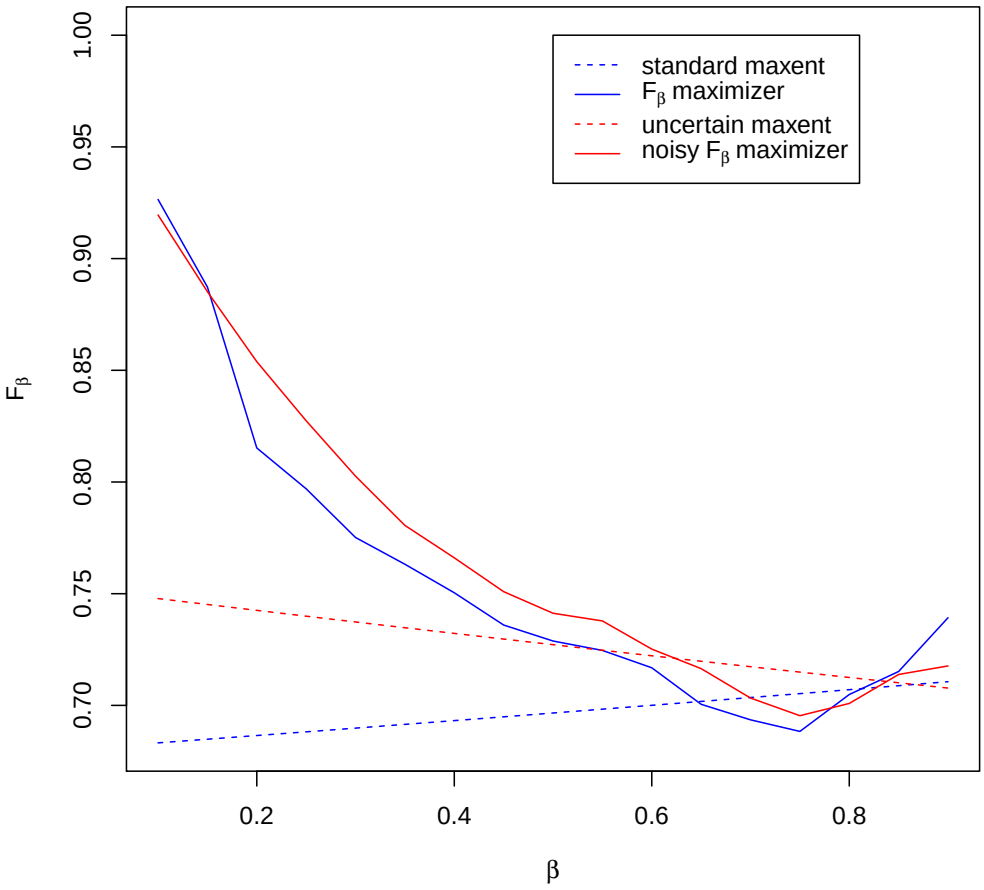

Figure 7: Average F_β on 1000 resamples from q on synthetic data B, with F_β^U maximizer fitted with weights chosen via deterministic F_β plug-in algorithm.

In Figure 8, we give a typical learning curve for the F_β^U maximizing algorithm. It demonstrates that the maximum F_β^U is reached quite quickly, typically for our examples in about 20 iterations.

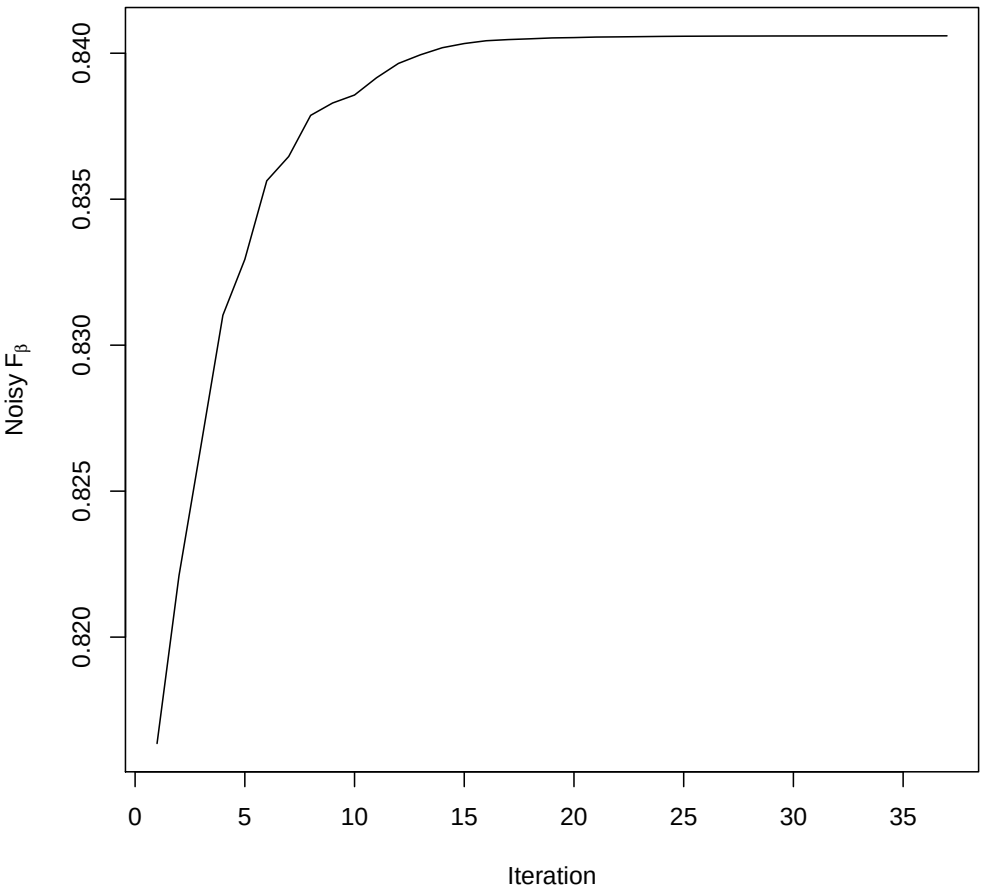

Figure 8: Typical learning curve for F_β^U maximization, here given for $\beta = 0.4$ on the dataset B.

In Figure 9, we give also a plot comparing the F_β^U curve for two different values of the `maxit` parameter in the logistic regression implemented in R. This parameter controls the maximum number of iterations in the likelihood maximization algorithm employed by R. The plot demonstrates that using only one step in their algorithm, which our algorithm guarantees to find a local maximum, does well, but not as well as using a much larger number of maximum iterations (`maxit=25`). That is, by doing a number of gradient ascent iterations towards the maximum of each weighted uncertain likelihood at every step of our algorithm, we are ending up with not only the closest local maximum, but indeed a better local maximum than we were guaranteed to find by doing only one step, and perhaps even a global maximum. It is a promising topic of future research to explore optimal schedules for the values of the maxit parameter that result in finding better maxima at possibly

the lowest computational cost. Borrowing some intuition from simulated annealing, one would start with a rather large value for maxit and gradually decrease it to one as the objective settles towards its maximum.

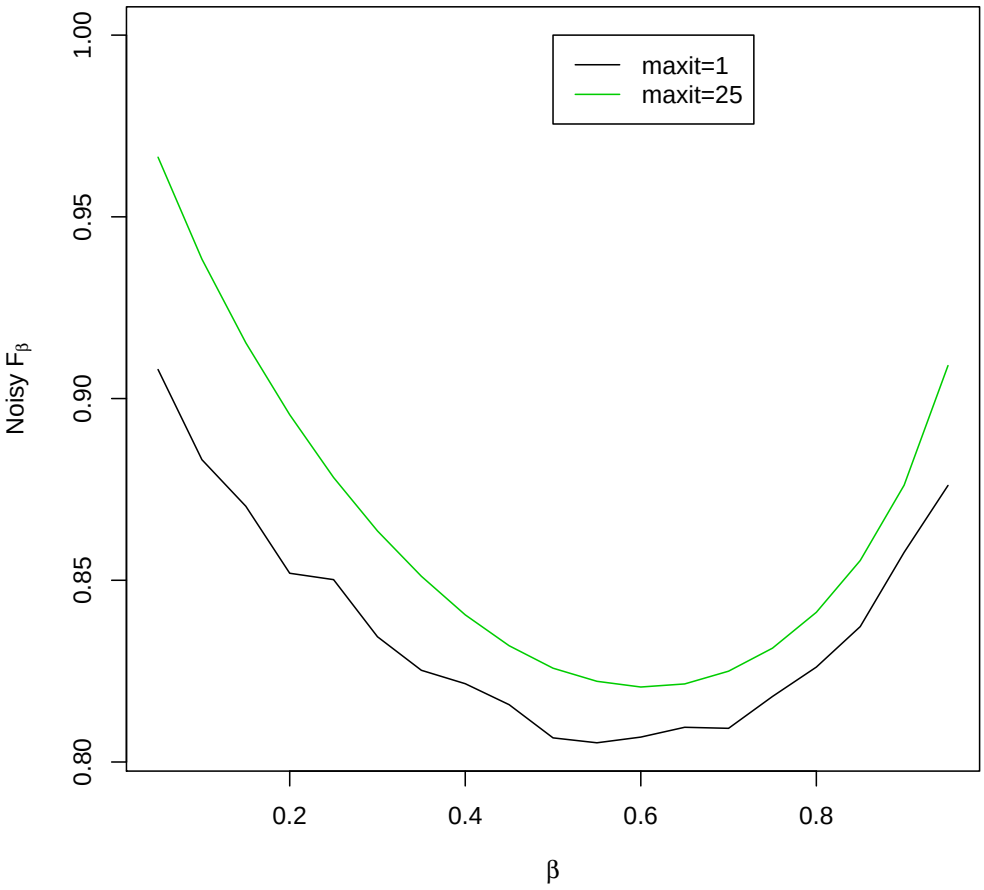

Figure 9: Comparative F_β^U curves for 1 and 25 steps of uncertain likelihood maximization at each iteration of the F_β^U maximizing algorithm, here given on the dataset B.

In Figure 10, we compare the performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the performance of the F-measure maximizing deterministic classifier and the noisy F-measure maximizing uncertain classifier on the Titanic data. In Figure 11, we examine the predictive performance of the noisy F-measure maximizing algorithm, comparing it with the baseline unweighted uncertain entropy model. In Figure 12, we compare the performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the performance of the F-measure maximizing deterministic classifier and the noisy F-measure

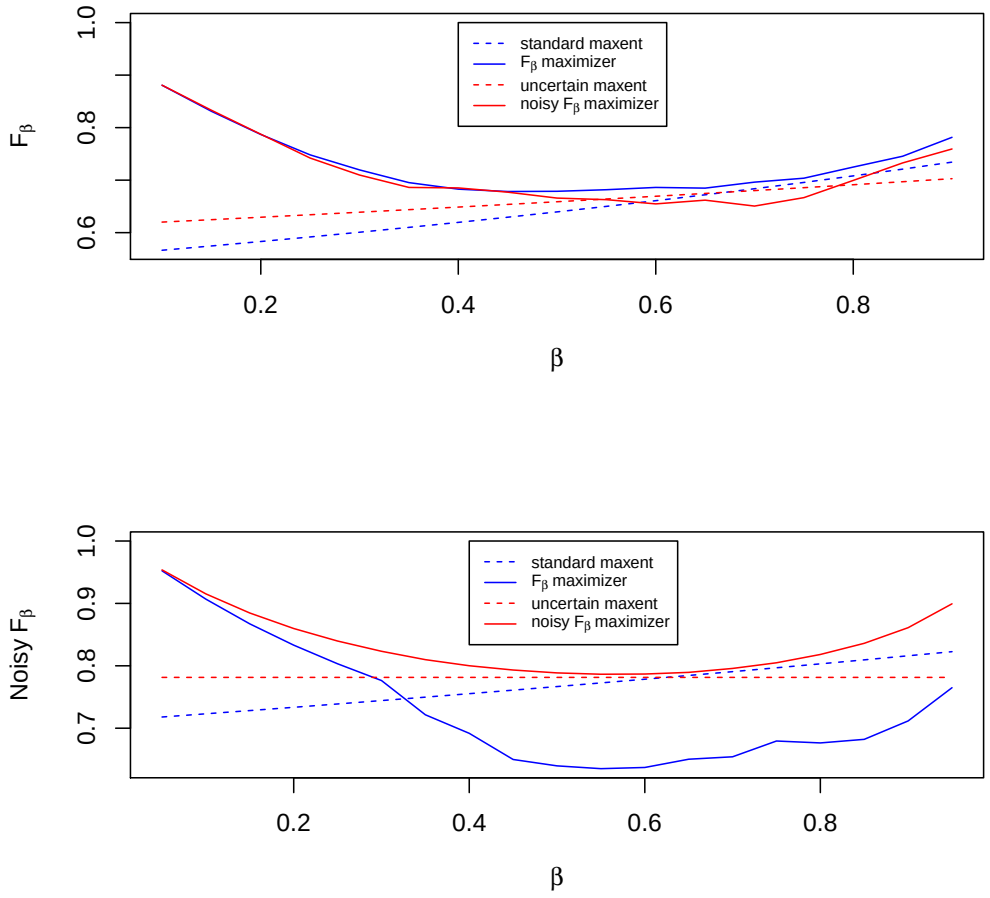

Figure 10: F_β and F_β^U on the training set for Titanic data.

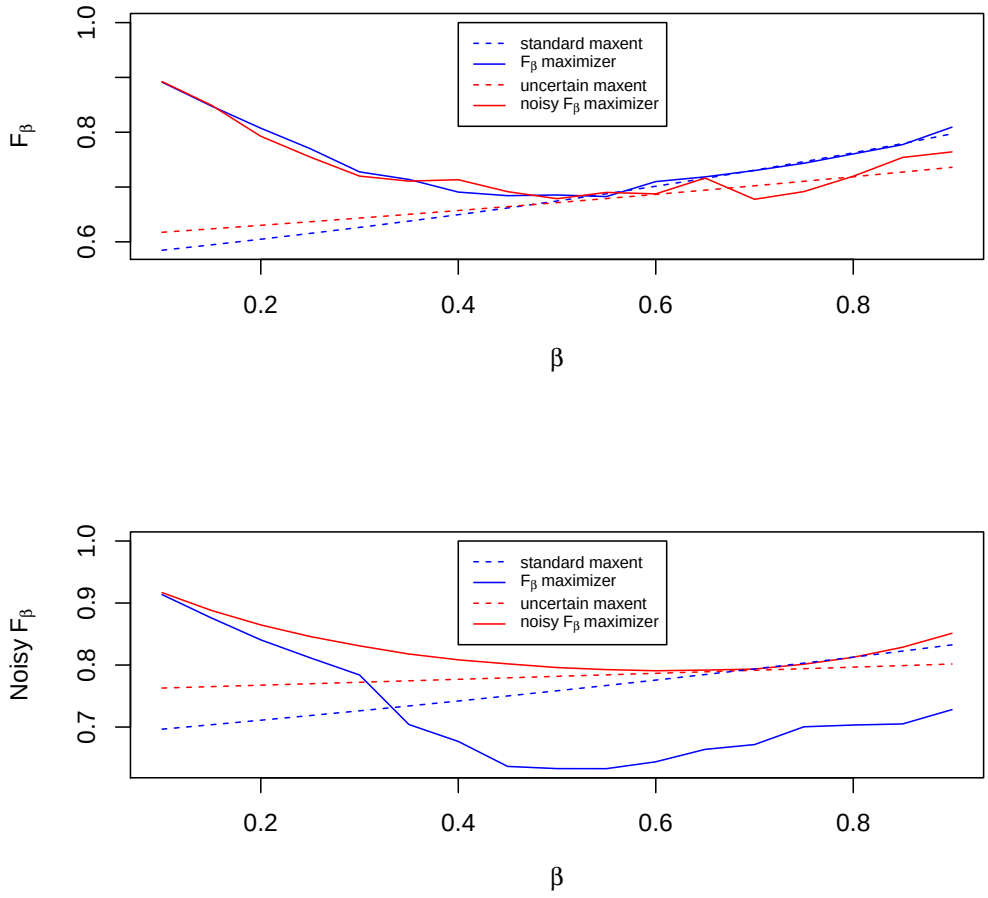

Figure 11: F_β^U on the test set for Titanic data.

maximizing uncertain classifier on the Titanic data with one-sided class noise. In Figure 13, we

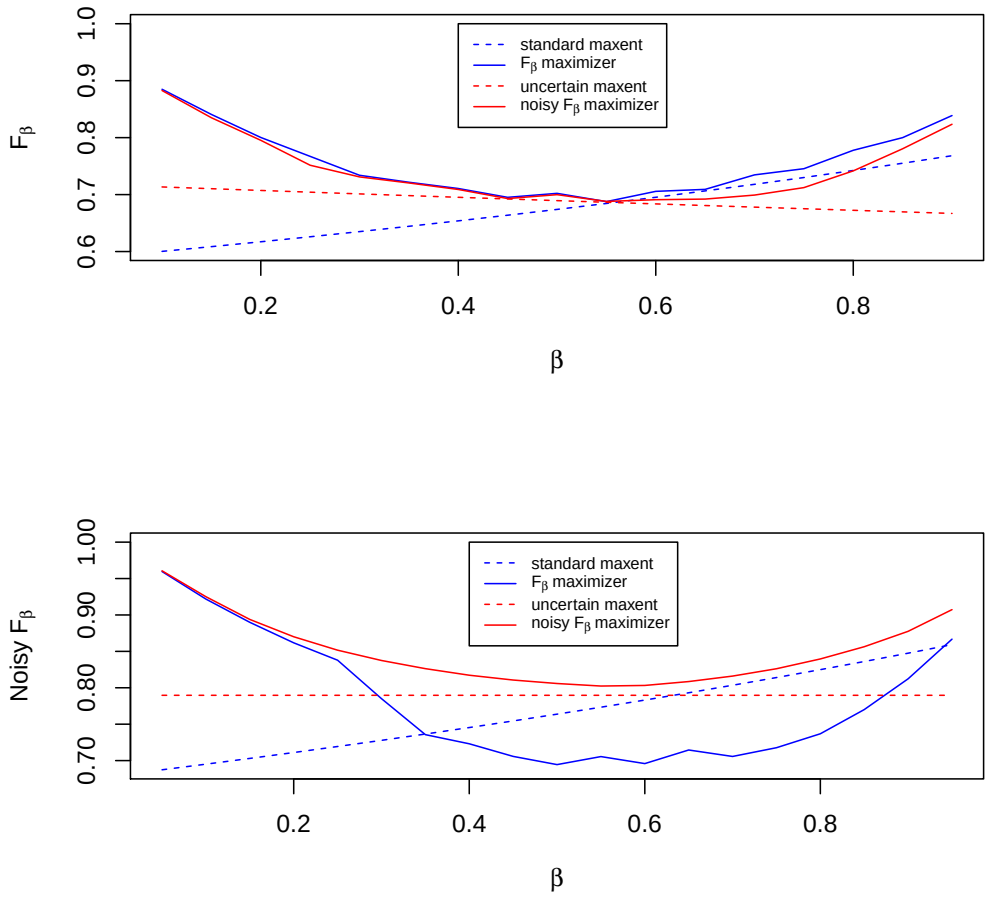

Figure 12: F_β and F_β^U on the training set for Titanic data with one-sided class noise.

examine the predictive performance of the noisy F-measure maximizing algorithm, comparing it with the baseline unweighted uncertain entropy model.

In Figure 14, we compare the performance (F-measure and noisy F-measure) of standard deterministic and uncertain classifiers with the performance of the F-measure maximizing deterministic classifier and the noisy F-measure maximizing uncertain classifier on the SPECT data. In Figure 15, we examine the predictive performance of the noisy F-measure maximizing algorithm, comparing it with the baseline unweighted uncertain entropy model.

The conclusions for our results in the case of the Titanic and SPECT data are similar to those we made for data A and B, indicating that our algorithm is ready to be employed in solving real-world problems. However, note that test set F_β^U for our algorithm has a tendency to sometimes fall below its baseline in datasets such as the Titanic data with one-sided noise and the SPECT data. This slight overfitting is due to a deficiency of data and again motivates us considering an extension of

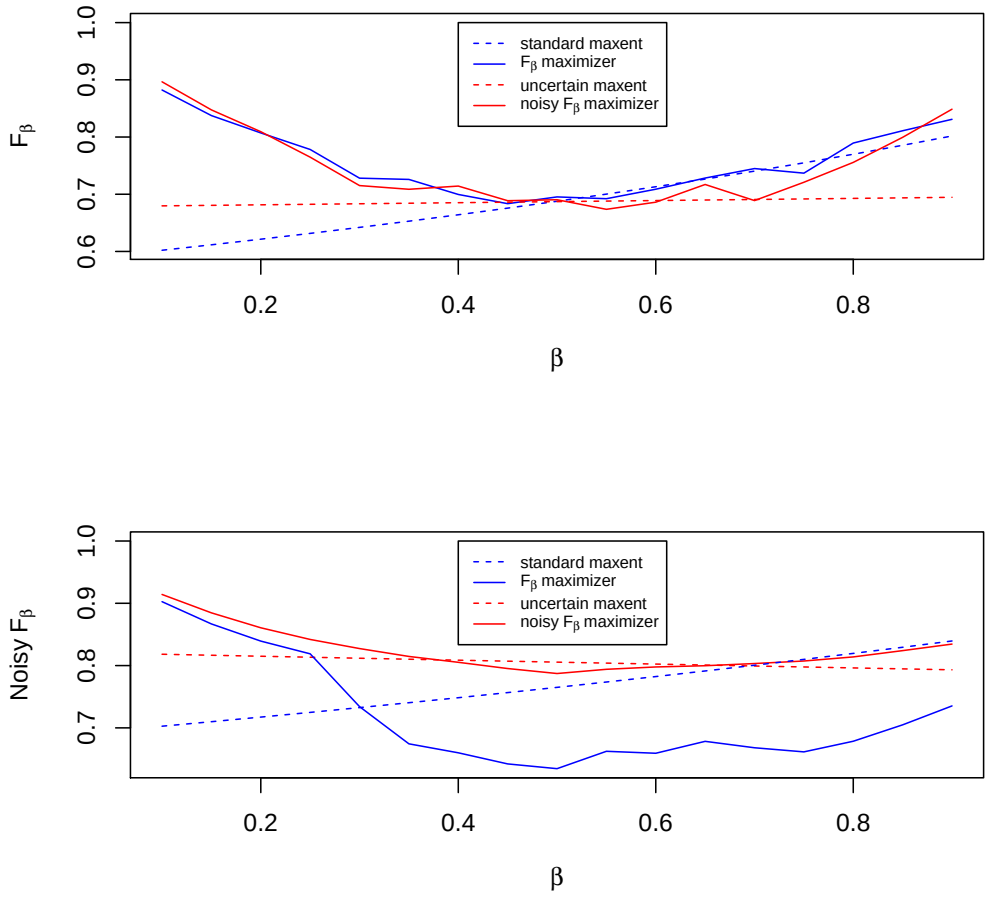

Figure 13: F_β and F_β^U on the test set for Titanic data with one-sided class noise.

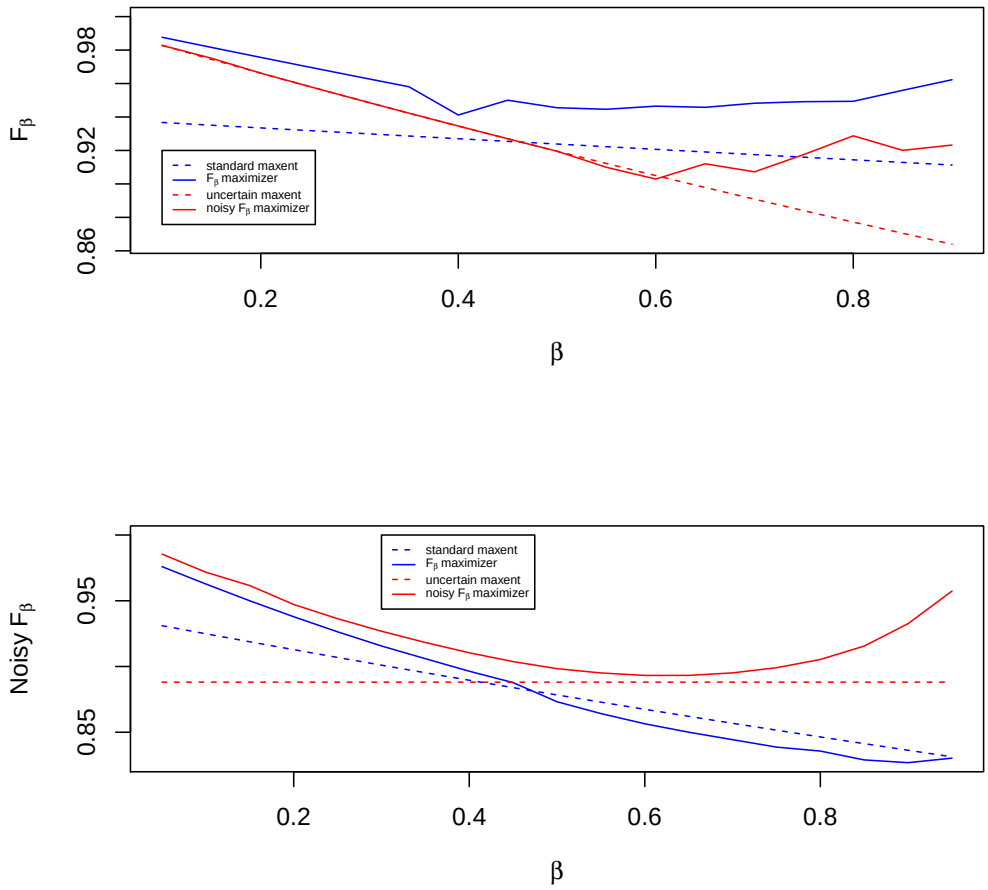

Figure 14: F_β and F_β^U on the training set for SPECT data.

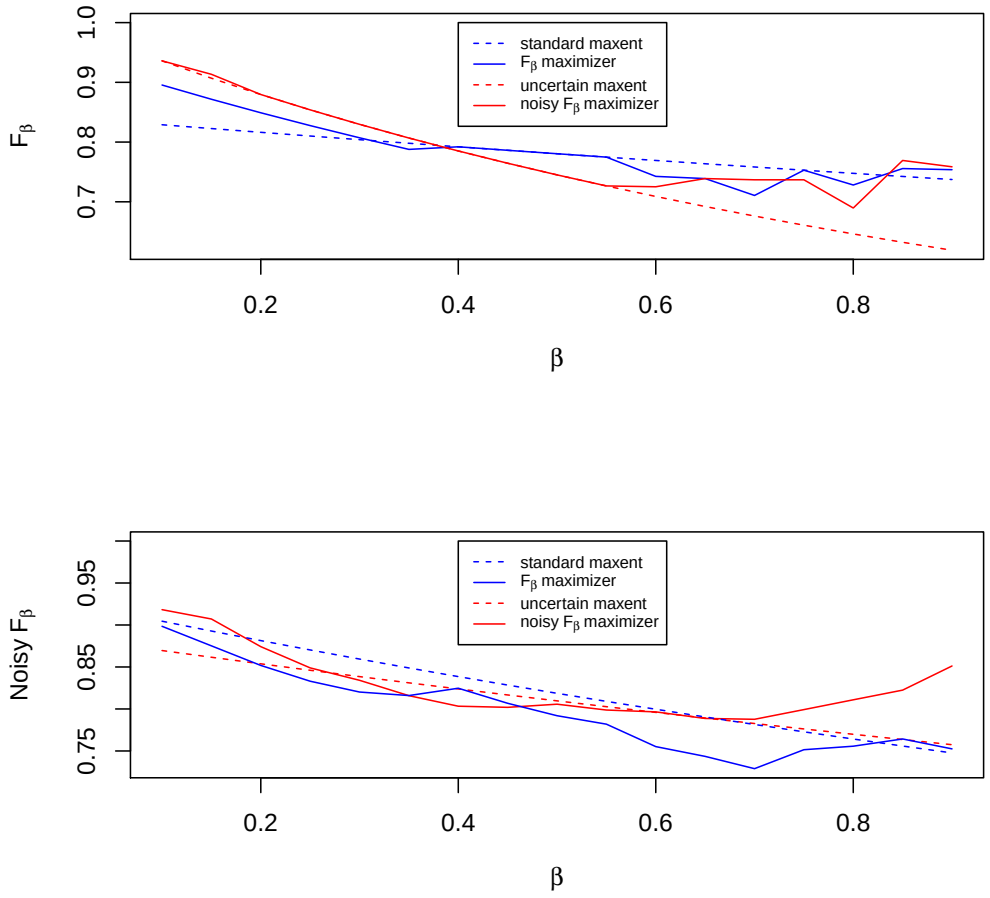

Figure 15: F_β and F_β^U on the test set for SPECT data.

our methodology to a regularized uncertain likelihood that will enable us to ensure that at all times we are minimizing overfitting.

7. Conclusion and future work

We have seen that in many situations, the training and test data might not be assigned with deterministic labels but rather with a distribution over the set of labels. This can be the case either due to intrinsic uncertainty (for example voting or "crowd labeling") of the examples, or due to observation noise. We consider a setup for handling such situations based on a maximum entropy model properly generalized to this type of data. We also propose a framework for measuring the performance of the classifier in the presence of uncertainty and based on it we pick a particular generalization of the celebrated F-measure, which is the noisy F-measure. In our point of view the proposed noisy F-measure has an intuitive interpretation and is a very good candidate for an alternative precision and recall balancing measure in the uncertain situation.

For the optimization of the noisy F-measure, we took the foundation introduced by Dimitroff et al. (2014), which offers a deterministic F-measure maximizing algorithm via a weighted maximum likelihood. We extended this framework to the uncertain setting, where classes are not provided deterministically, but instead there exist probabilities that an example belongs to each class. Optimizing the noisy F-measure involves minimizing average Kullback-Leibler divergences between true and model distributions, which we showed can be performed via a maximum weighted uncertain likelihood, and we provided experimental evidence on several data sets for the resulting algorithm's success. Apart from the experimental results we also gave a theoretical justification for the the performance of the algorithm.

Having said this, we also point out that there is much scope for future work. There is the immediate question of how to choose a dynamic schedule for the number of steps (the `maxit` parameter) in the uncertain likelihood maximization performed at each iteration of our noisy F-measure maximizing algorithm, in order ideally to reach a global maximum. An optimal schedule would possibly have implications in a much broader context than the uncertain likelihood and F-measure maximizer discussed in this paper. In order to make the (noisy) F-measure fully "operational" we must add regularization and correspondingly generalize the results and the algorithm. In an upcoming work we consider a generalization of the deterministic F-measure maximizing algorithm of Dimitroff et al. (2014) to the regularized and multiclass cases which should lead to a similar natural extension of the noisy F-measure maximizing algorithm. In the multiclass setup, the data would be a set of examples with labels being discrete distributions over the set of possible classes. The regularization boils down to maximizing a penalized (noisy) F measure with the help of a regularized weighted (uncertain) likelihood.

We additionally note that in setting up the noisy F-measure we have chosen to parameterize the dependence between true and modeled classes in a very specific way. However, as we mentioned, there do exist other ways of representing the structure of dependence between experimental and model class distributions. This warrants further investigation, in particular to answer the question of whether there is ever a "best" such formulation.

There is also the interesting question of how this setup translates for different models, in particular SVM where there is no direct probabilistic interpretation of the classifier.

A major topic for further research is to adapt and apply the model to semi-supervised learning tasks, with an appealing link to the generalized expectation criteria for learning from weakly labeled

data as discussed by Mann and McCallum (2010). The uncertain training examples in our setup can be viewed as weakly labeled data and the corresponding uncertain likelihood and noisy F-measure optimization can be applied. Also, questions about the uncertainty in the data and whether it can be re-assessed by looking at the distance between prior and posterior model distributions could have major practical relevance.

On the theoretical side, it would be appealing to establish a precise link to Bayesian models. In our setup, the observations themselves have prior distributions, not the parameters of the model as it is in the Bayesian approach. However the uncertainty of the observations does imply uncertainty of sufficient statistics and the parameter models, so the link at least on an intuitive level is very clear.

Acknowledgments

The work described in this paper is supported by the FP7-ICT Strategic Targeted Research Project PHEME (No. 611233).

A. Bounding the weights

We justify bounding the weights theoretically in the following way. The idea is to smooth out the confusion counts in order to avoid the use of indicator functions in (4.8), replacing them with expressions which are linear in $p(c_1|\, x_i, \hat{\lambda}_\beta) - q_{i1}$ close to the singularity. When these terms cancel with the problematic denominator, we are left with bounded weights.

Consider smoothing TP^U:

$$\begin{aligned} TP^U &= \sum_{i=1}^{m} \min(q_{i1}, p(c_1|\, x_i, \lambda)) \\ &= \sum_{i=1}^{m} \left[p(c_1|\, x_i, \lambda) - \max(0, p(c_1|\, x_i, \lambda) - q_{i1})\right] \\ &\approx \sum_{i=1}^{m} \left[p(c_1|\, x_i, \lambda) - g_1(p(c_1|\, x_i, \lambda) - q_{i1})\right]. \end{aligned}$$

And similarly,

$$TN^U \approx \sum_{i=1}^{m} \left[q_{i0} - g_2(p(c_1|\, x_i, \lambda) - q_{i1})\right],$$

for some choice of smooth functions g_1 and g_2 that approximate $\max(0, x)$.

Then we have:

$$\begin{aligned} \frac{\partial TP^U}{\partial p(c_1|\, x_i, \lambda)} &= 1 - g_1'(p(c_1|\, x_i, \lambda) - q_{i1}) \\ \frac{\partial TN^U}{\partial p(c_1|\, x_i, \lambda)} &= -g_2'(p(c_1|\, x_i, \lambda) - q_{i1}). \end{aligned}$$

And so,

$$w(\beta)_i = \frac{p(c_1|\, x_i, \hat{\lambda}_\beta) \cdot p(c_0|\, x_i, \hat{\lambda}_\beta)}{p(c_1|\, x_i, \hat{\lambda}_\beta) - q_{i1}} \cdot \frac{F^U_\beta(\hat{\lambda}_\beta)}{TP^U(\hat{\lambda}_\beta)} \Big[\beta F^U_\beta(\hat{\lambda}_\beta) g_2' \Big(p(c_1|\, x_i, \hat{\lambda}_\beta) - q_{i1}\Big) - (1 - \beta F^U_\beta(\hat{\lambda}_\beta))(1 - g_1' \Big(p(c_1|\, x_i, \hat{\lambda}_\beta) - q_{i1}\Big)\Big]. \quad \text{(A.1)}$$

One appropriate choice of functions g_1 and g_2 is the following. Suppose $\epsilon > 0$ and let:

$$f_1(x) = \frac{(x+\epsilon)^2}{2\epsilon}$$
$$f_2(x) = \frac{x^2}{2\epsilon}.$$

Now set:

$$g_1(x) = \begin{cases} 0, & \text{if } x \leq -\epsilon. \\ f_1(x), & \text{if } -\epsilon < x \leq 0. \\ x + \frac{\epsilon}{2}, & \text{if } x > 0. \end{cases} \quad \text{(A.2)}$$

$$g_2(x) = \begin{cases} 0, & \text{if } x \leq 0. \\ f_2(x), & \text{if } 0 < x < \epsilon. \\ x - \frac{\epsilon}{2}, & \text{if } x \geq \epsilon. \end{cases} \quad \text{(A.3)}$$

Then we find that for $p(c_1|\, x_i, \lambda) - q_{i1} \in [-\epsilon, 0]$,

$$w(\beta)_i = \frac{p(c_1|\, x_i, \hat{\lambda}_\beta) \cdot p(c_0|\, x_i, \hat{\lambda}_\beta) \cdot F^U_\beta(\hat{\lambda}_\beta)(1 - \beta F^U_\beta(\hat{\lambda}_\beta))}{\epsilon TP^U(\hat{\lambda}_\beta)}.$$

Similarly for $p(c_1|\, x_i, \lambda) - q_{i1} \in [0, \epsilon]$,

$$w(\beta)_i = \frac{p(c_1|\, x_i, \hat{\lambda}_\beta) \cdot p(c_0|\, x_i, \hat{\lambda}_\beta) \cdot \beta (F^U_\beta)^2(\hat{\lambda}_\beta)}{\epsilon TP^U(\hat{\lambda}_\beta)}.$$

These cases taken together show that as $p(c_1|\, x_i, \lambda) - q_{i1} \to 0$, the weights are bounded in inverse proportionality to ϵ.

Hence we have shown that the bounds on the weights in the algorithm correspond to a slight smoothing of noisy TP and TN counts.

References

C. C. Aggarwal. Managing and mining uncertain data. *ser. Advances in database systems*, (5), 2009. 3

C. C. Aggarwal and P. S. Yu. A survey of uncertain data algorithms and applications. *IEEE Transactions on Knowledge and Data Engineering*, 21(5):609–623, 2009. 3

K. Bache and M. Lichman. UCI machine learning repository, 2013. URL `https://archive.ics.uci.edu/ml/datasets/SPECT+Heart`. 15

A.L. Berger, V.J. Della Pietra, and S.A. Della Pietra. A maximum entropy approach to natural language processing. *Comput. Linguist.*, 22(1):39–71, 1996. 4

T. Cohn and L. Specia. Modelling annotator bias with multi-task Gaussian processes: an application to machine translation quality estimation. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL '13)*, pages 32–42, Sofia, Bulgaria, 2013. Association for Computational Linguistics. 2, 3

Krzysztof Dembczyński, Willem Waegeman, Weiwei Cheng, and Eyke Hü llermeier. An exact algorithm for F-measure maximization. In *Neural information processing systems : 2011 conference book*. Neural Information Processing Systems Foundation, 2011. 4

A.P. Dempster, N.M. Laird, and D.B. Rubin. Maximum likelihood from incomplete data via the EM algorithm. *Journal of the Royal Statistical Society. Series B (Methodological)*, 39(1):1–38, 1977. 3

Thierry Denoeux. Maximum likelihood estimation from uncertain data in the belief function framework. *IEEE Trans. Knowl. Data Eng.*, 25(1):119–130, 2013. 3

Georgi Dimitroff, Laura Toloşi, Borislav Popov, and Georgi Georgiev. Weighted maximum likelihood for optimization of the F-measure of a maximum entropy classifier, 2013. http://www.ontotext.com/publications/2012. 2

Georgi Dimitroff, Georgi Georgiev, Laura Tolosi, and Borislav Popov. Efficient F-measure maximization via weighted maximum likelihood. *Machine Learning*, 2014. ISSN 0885-6125. 3, 4, 7, 32

M. Ehrgott. *Multicriteria Optimization*. Springer, New Jersey, 2005. 11

Charles Elkan and Keith Noto. Learning classifiers from only positive and unlabeled data. In *Proceedings of the 14th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '08, pages 213–220, New York, NY, USA, 2008. ACM. 3

M. Jansche. Maximum expected F-measure training of logistic regression models. In *HLT '05*, pages 692–699, Morristown, NJ, USA, 2005. Association for Computational Linguistics. 3, 4

H.-P. Kriegel and M. Pfeifle. Density-based clustering of uncertain data. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining. Chicago, Illinois, USA: ACM*, pages 672–677, 2005. 3

G.S. Mann and A. McCallum. Generalized expectation criteria for semi-supervised learning with weakly labeled data. *Journal of Machine Learning Research*, 11:955–984, 2010. 2, 33

Naresh Manwani and P. S. Sastry. Noise tolerance under risk minimization. *CoRR*, abs/1109.5231, 2011. 3

Ye Nan, Kian Ming Adam Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing F-measure: a tale of two approaches. In *ICML*, 2012. URL `http://dblp.uni-trier.de/db/conf/icml/icml2012.html#NanCLC12`. 4

Nagarajan Natarajan, Inderjit Dhillon, Pradeep Ravikumar, and Ambuj Tewari. Learning with noisy labels. In C.J.C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K.Q. Weinberger, editors, *Advances in Neural Information Processing Systems 26*, pages 1196–1204. Curran Associates, Inc., 2013. 3

R.M. Neal and G.E. Hinton. A view of the EM algorithm that justifies incremental, sparse and other variants. *Learning in Graphical Models*, pages 355–368, 1998. 6

David M. W. Powers. Evaluation: From Precision, Recall and F-Factor to ROC, Informedness, Markedness & Correlation. *Journal of Machine Learning Technologies*, 2(1):37–63, 2011. 4

V.C. Raykar, S. Yu, L.H. Zhao, G.H. Valadez, C. Florin, L. Bogoni, and L. Moy. Learning from crowds. *Journal of Machine Learning Research*, 11:1297–1322, 2010. 3

K. Yip W. Ho S. Tsang, B. Kao and S. Lee. Decision trees for uncertain data. *IEEE Transactions on Knowledge and Data Engineering*, 23(1):64–78, 2011. 3

Clayton Scott, Gilles Blanchard, and Gregory Handy. Classification with asymmetric label noise: consistency and maximal denoising. *JMLR W&CP*, 30:489–511, 2013. 3

C. van Rijsbergen. *Information Retrieval*. Butterworths, London, 1979. 7

Vanderbilt University. Vanderbilt Biostatistics Datasets, 2014. URL `http://biostat.mc.vanderbilt.edu/wiki/Main/DataSets`. 14

C. K. Chui R. Cheng M. Chau W. K. Ngai, B. Kao and K. Y. Yip. Efficient clustering of uncertain data. In *Sixth International Conference on Data Mining (ICDM '06)*, pages 436–445, 2006. 3