# Weighted maximum likelihood as a convenient shortcut to optimize the F-measure of maximum entropy classifiers


## Abstract

We link the weighted maximum entropy and the optimization of the expected $F_\beta$-measure, by viewing them in the framework of a general common multi-criteria optimization problem. As a result, each solution of the expected $F_\beta$-measure maximization can be realized as a weighted maximum likelihood solution - a well understood and behaved problem. The specific structure of maximum entropy models allows us to approximate this characterization via the much simpler class-wise weighted maximum likelihood. Our approach reveals any probabilistic learning scheme as a specific trade-off between different objectives and provides the framework to link it to the expected $F_\beta$-measure.


## 1 Introduction

In many NLP classification applications, the classes are not symmetric and the user has some preference towards a high Precision or Recall of a particular target class. Thus, appropriate tuning of the model is often necessary, depending on the particular tolerance of the application to false positive or false negative results. This preference can be expressed by requiring a large $F_\beta$ measure for a particular $\beta$ describing the desired Precision/Recall trade-off. Ideally, the parameters of the linear model should be estimated such that a desired $F_\beta$ measure is maximized. However, directly maximizing $F_\beta$ is hard, due to its non-concave shape.

Maximum likelihood-based classifiers such as the maximum entropy are relatively easy to fit, but they are rigid and cannot be tuned to a desired Precision and Recall trade-off. In this article, we consider a more flexible maximum entropy model, which optimizes a *weighted* likelihood function. If appropriate weights are chosen, then the maximum weighted likelihood model coincides with the optimal $F_\beta$ model. The advantage of the weighted likelihood as a loss function is that it is concave and standard gradient methods can be used for its optimization. In fact an existing maximum entropy implementation can be easily generalized to the weighted case.

To the best of our knowledge, such a link between the maximum likelihood and the $F_\beta$ has not been established before. The article is focused on the intuition of the relation and the sketch of the proof of the main result. We also present numerical experiment supporting the theoretical findings. Additional value of our theoretical observation is that it establishes the methodology of viewing a particular probabilistic model as a specific solution of a common multi-criteria optimization problem.

This article is organized as follows. In Section 2 we present related work, Sections 3 to 6 present the theoretical aspects of link between the weighted maxent and $F$ measure. Section 7 introduces the algorithm, Section 8 explains the steps for evaluation of the algorithm, Section 9 presents the datasets. Sections 10 and 11 present aspects of performance of our method on the datasets and Section 12 concludes the paper.

## 2 Related work

The most popular heuristic for Precision-Recall trade-off is based on adjusting the acceptance threshold given by maximum entropy models (or

any learning framework). However, this procedure amounts to a simple translation of the maximum likelihood hyperplane towards or away from the target class and does not fit the model anew.

The expected $F$ measure $\tilde{F}$ is also considered in (Nan et al., 2012), where also its consistency is studied and even a Hoeffding bound for the convergence is given. However, the authors there mainly concentrate on the acceptance threshold to optimize the $F$-measure.

(Dembczyn'ski et al., 2011) gave a general algorithm for $F$ measure optimization for a particular parametrization involving $m^2 + 1$ parameters where $m$ is the number of examples in the binary classification case. Determining the parameters of the models however can be very hard. A very interesting result in (Dembczyn'ski et al., 2011) is that in the worst case there is a lower bound on the discrepancy between the optimal solution and the solution obtained by means of optimal acceptance threshold, which further motivates our approach. In our approach we directly find the parameters of the model that maximize the expected $F$ measure using the link to the weighted maximum likelihood.

(Jansche, 2005) describe a maximum entropy model that optimizes directly an expected $F_\beta$-based loss. However the expected $F_\beta$ is not concave and is rather cumbersome to deal with. Therefore the standard gradient methods do not guarantee optimality of the solution.

(Minkov et al., 2006) introduce another heuristics, which is based on changing the weight of a special feature, which indicates if a sample is in the complementary class or not.

The weighted logistic regression is well known, see for example (Vandev and Neykov, 1998), and the corresponding estimation is barely harder than in the standard case without weights. See also (Simecková, 2005) for an interesting discussion.

## 3 The Maximum Entropy Model

The maximum entropy modeling framework as introduced in the NLP domain by (Berger et al., 1996) has become the standard for various NLP tasks. To fix notations consider a training set of $m$ samples $\{(x(i), y(i)) : i \in 1, \dots m\}$ where $x(i)$ is a sample with class $y(i)$, where $y(i)$ takes values in some finite set $\mathcal{Y}$. In this paper we aim at explaining the main idea of the link between the weighted maximum entropy and the expected $F_\beta$; to keep things technically simple we restrict to the case $|\mathcal{Y}| = 2$. Each observation is represented by a set of features $\{f_j(x(i), y(i)) : j \in 1, \dots, N\}$.

The maximum entropy principle forces the model conditional probabilities $p(y|x, \lambda)$ to have the form:

$$p(y|x, \lambda) = \frac{1}{Z_\lambda(x)} \exp \sum_j \lambda_j \cdot f_j(x, y),$$

where $\lambda \in \mathbb{R}^N$ are the model parameters and $Z_\lambda(x)$ is a normalization constant. The calibration of the model amounts to (see (Berger et al., 1996)) maximizing the log-likelihood

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

In the following for a weight vector $w \in \mathbb{R}^m$ we will make use of the weighted log-likelihood function

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

In our case the weights will be defined mostly class-wise, i.e. examples from the same class will always have the same weights.

## 4 Precision/Recall trade off. Expected $F_\beta$-measure.

The performance of a classifier is typically measured using the Precision and Recall metrics, and in particular their tradeoff described by a constant $\beta \in [0, 1]$ and expressed as the $\beta$-weighted harmonic mean called $F_\beta$-measure:

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

The larger the $\beta$ the greater the influence of the Precision as compared to the Recall on the $F_\beta$-measure. The Precision and Recall are defined in terms of the true/false positive/negative counts.

For a given example with attributes $x$ the maximum entropy model will produce the conditional probabilities $p(y|x, \lambda)$ of the example being into one of the classes $y \in \mathcal{Y}$. When used for classification however, one would typically choose the class $y(x)$ having the largest probability i.e.

$$y(x) = \text{argmax}_y p(y|x, \lambda).$$

This means that we would completely disregard the additional information incorporated into the model. A more probabilistic approach would be to draw the class $y(x)$ randomly out of the model distribution given by the probability weights $\{p(y|x,\lambda) : y \in \mathcal{Y}\}$. This way the classes $y(x)$ as well as the true/false positive/negative counts would be random variables. However if we perform this sampling many times and take the average we will end up having the expected true/false positive/negative counts. For example the expected true positive and true negative counts are given by

$$\begin{aligned} \tilde{A}_u &= \mathbb{E}\#\text{true pos} = \sum_{i:y(i)=1} p(1|x(i),\lambda); \\ \tilde{D}_u &= \mathbb{E}\#\text{true neg} = \sum_{i:y(i)=0} p(0|x(i),\lambda) \end{aligned} \tag{1}$$

Using the expected counts instead of the realized ones we can define the mean field approximation $\tilde{P}$ and $\tilde{R}$ of the precision and recall metrics and consequently define the mean field approximation $\tilde{F}_\beta$ of the standard $F_\beta$ measure

$$\tilde{F}_\beta := \left( \frac{\beta}{\tilde{P}} + \frac{1-\beta}{\tilde{R}} \right)^{-1}.$$

As in (Jansche, 2005) with a slight abuse of notation we will call $\tilde{F}_\beta$ the expected $F_\beta$ measure. For a large training set and a good model the expected $F_\beta$ measure on the training set will be close to the standard one since the model probabilities $p(y(i)|x(i),\lambda)$ will be close to one for the training examples.

## 5 Weighted maximum likelihood vs. expected $F_\beta$-measure maximization.

Clearly the log-likelihood and the expected $F_\beta$ measure are two different, however one would hope, not orthogonal objectives.

Intuitively every reasonable machine learning model would try to set the model parameters $\lambda$ in such a manner that for all training examples the model conditional probabilities of the observed classes $y(i)$ given the example's attributes $x(i)$, namely $p(y(i)|x(i),\lambda)$, are as large as possible. In general if the used model is not overfitting it would not be possible for all conditional probabilities to be close to one simultaneously, and implicitly every particular model would handle the trade-offs in its own manner. In this sense the important difference between the log-likelihood and the expected $F_\beta$ measure seen as objective functions is that, while the log-likelihood approach gives equal importance to all training examples on the logarithmic scale the (expected) $F_\beta$ measure has a parameter controlling this trade-off on a class-wise level. On the other hand, as noted in (Jansche, 2005) the flexibility in $\tilde{F}_\beta$ comes at a price - the $\tilde{F}_\beta$ is by far not that nice function to optimize as the log-likelihood is. The next proposition gives a useful link between the $\tilde{F}_\beta$ and the weighted log-likelihood enabling us to find $\tilde{F}_\beta$ optimizers by solving the very well behaved and understood weighted maximum likelihood problem.

**Proposition 1.** *Let $\hat{\lambda}_\beta$ be the maximizer of the expected $F_\beta$ measure $\tilde{F}_\beta$. Then there exists a vector of weights $w(\beta) \in \mathbb{R}^m$ such that $\hat{\lambda}_\beta$ coincides with the weighted maximum likelihood estimator*

$$\hat{\lambda}_{ML}^{w(\beta)} = arg\max l^W(\lambda : w(\beta), x, y)$$

*Moreover, we can approximate the $\beta$-implied weights $w(\beta)$ with a class-wise weight vector $\bar{w}(\beta)$ (i.e., the weights of training examples from the same class have the same weights) , that is*

$$\hat{\lambda}_\beta = \hat{\lambda}_{ML}^{w(\beta)} \quad \textit{and} \quad \hat{\lambda}_\beta \approx \hat{\lambda}_{ML}^{\bar{w}(\beta)}$$

Below we give the intuition of the proof and some formal arguments, without presenting all technical details, due to lack of space.

**Sketch of proof**:

The proof makes use of multicriteria optimization techniques (Ehrgott, 2005), which are typically applied when two or more conflicting objectives need to be optimized simultaneously. In our case, the number of true positives and the number of true negatives need to be maximized at the same time, but most classifiers (at least those that do not overfit badly) trade-off between them. The solutions of multicriteria optimization problem are called Pareto optimal solutions. A solution is Pareto optimal if none of the objectives can be improved without deteriorating at least one of the other objectives.

Intuitively, the maximum likelihood optimizes simultaneously the conditional probabilities $p(y(i)|x(i),\lambda)$ via implicitly setting some trade-offs between them. Therefore our idea is to adjust these trade-offs using the weights in such a manner that the $\tilde{F}_\beta$ is optimized rather than the likelihood. The most natural and general

way to look at these trade-offs is to consider the multicriteria optimization problem (MOP) $\max\{\log p(y(1)|x(1),\lambda),...,\log p(y(m)|x(m),\lambda)\}$. It turns out that both the max likelihood and the $\tilde{F}_\beta$ optimizer are particular solutions of the MOP. On the other hand all solutions of the MOP can be obtained by maximizing nonnegative linear combinations of the objectives (Ehrgott, 2005). However a nonnegative combination of the objectives $\{\log p(y(1)|x(1),\lambda),...,\log p(y(m)|x(m),\lambda\}$ is precisely the weighted maximum entropy objective function.

Technically, for each $\beta$ the $\tilde{F}_\beta$ maximizer $\hat{\lambda}_\beta$ can actually be seen as an element of the Pareto optimal set of the multi-criteria optimization problem

$$\max_\lambda\{\tilde{A}(\lambda),\tilde{D}(\lambda)\}, \tag{2}$$

where $\tilde{A}(\lambda)$ and $\tilde{D}(\lambda)$ are the model expected true positive and true negative counts on the training set. This follows from the fact that we can rewrite $\tilde{F}_\beta$ as follows:

$$\tilde{F}_\beta(\lambda)=\frac{\tilde{A}(\lambda)}{\beta(\tilde{A}(\lambda)-\tilde{D}(\lambda))+(1-\beta)m_1+\beta m_0},$$

where $m_1$ is the total number of positive examples and $m_0$ the number of negative ones. Furthermore the Pareto optimal set of (2) is a subset of the Pareto optimal set of the finer granularity multi-criteria optimization problem

$$\max_\lambda\{p(y(1)|x(1),\lambda),...,p(y(m)|x(m),\lambda)\}.$$

Clearly, because of the strict monotonicity of the logarithm the above optimization problem is equivalent to

$$\max_\lambda\{\log p(y(1)|x(1),\lambda),...,\log p(y(m)|x(m),\lambda)\}. \tag{3}$$

On the other hand each element of the Pareto optimal set of (3) can be realized as a weighted maximum likelihood estimator associated to some weight vector $w \in R^m$, which concludes the proof. The pass to approximate class-wise weights is achieved using a linearization of the log-conditional probabilities of the training examples. □

## 6 Interpretation of the weights

Apart from the obvious technical generalization of the likelihood function the weights could on average be interpreted as a modification of the training set by adding new examples with intensity $w(i)$ while keeping the attributes and the classes $(x(i),y(i))$. In particular for $w(i)<1$ the $i$th example is deleted with probability $1-w(i)$. If $w(i)>1$, say $w(i)=q+w_f(i)$ for some integer $q\geq 1$ and $0\leq w_f(i)<1$ then generate $q$ identical training examples $(x(i),y(i))$ and additionally clone it with probability $w_f(i)$.

This view highlights yet another interpretation of the weights: an asymmetric regularization. Removing some examples when the weight is smaller than 1 is a well known regularization technique called drop-out. When it is applied to features involving only a subset of the classes then obviously it is an asymmetric regularization. The case of weights larger than 1 can be viewed in the same light by simple renormalization. If we have an exogenous $L^2$ regularization, adding class-wise weights would alter the influence of the regularization on the parameters corresponding to different classes, yet again we achieve an asymmetric regularization.

## 7 The algorithm

We search for a value $w$ in a predefined interval $[w_{min},w_{max}]$ which gives maximum $F_\beta(w)$. Our experiments on artificial and real data suggest that the expected $F_\beta(w)$ is unimodal on intervals like $[\varepsilon,w_{max}]$, for a small $\varepsilon$ close to zero. This suggests that a golden section search algorithm (Kiefer, 1953) can find the maximum efficiently, i.e. with a minimum number of trained weighted likelihood models.

In practice however the estimate of $F_\beta(w)$ may not be unimodal, because numerical methods are used for training weighted maximum entropy models and the optimal model is only approximately identified. It is safe to assume however that deviation from unimodality is not considerable, for example, we can accept that the function $F_\beta(w)$ is $\delta$ - unimodal (as defined in (Brent, 1973)) for some $\delta$. Then, (Brent, 1973) show that the golden section search approximates the location of the maximum with a tolerance of $5.236\delta$.

Below we describe the steps of the algorithm:

## 8 Evaluation of the algorithm

In order to demonstrate that our algorithm is an efficient tool for optimizing the $F_\beta$ measure, we performed the following tests, the results of which

**Algorithm 1** Golden Section Search

**Require:** Unimodal function $f$, interval $[a, b]$
**Ensure:** $x^* = \arg\max_x f(x)$

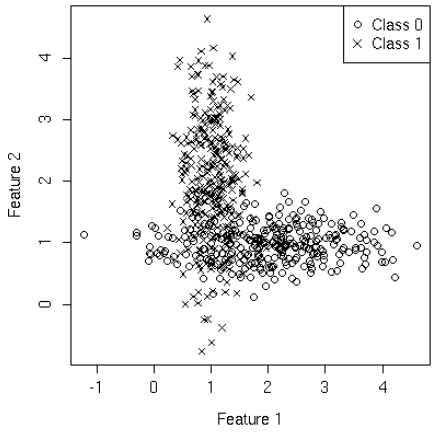
$\phi \leftarrow \frac{1+\sqrt{5}}{2}$
**function** GSS($f$, $a$, $b$, $p_1$, $p_2$)
  **if** $|b - a| < \varepsilon$ **then**
    **return** $a$
  **else**
    **if** $f(p_1) > f(p_2)$ **then**
      $b \leftarrow p_2$
      $p_2 \leftarrow p_1$
      $p_1 \leftarrow (2 - \phi)(b - a)$
    **else**
      $a \leftarrow p_1$
      $p_1 \leftarrow p_2$
      $p_2 \leftarrow (2 - \phi)(b - a)$
    **end if**
    **return** GSS($f, a, b, p_1, p_2$)
  **end if**
**end function**
$p_1 \leftarrow a + (2 - \phi)(b - a)$
$p_2 \leftarrow b - (2 - \phi)(b - a)$
$x^* \leftarrow GSS(f, a, b, p_1, p_2)$

are described in the Results section.

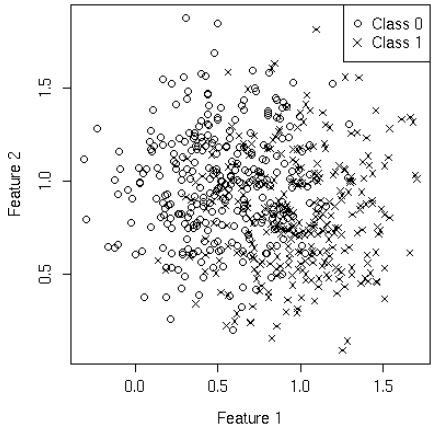
First, we evaluated Precision and Recall at different values of the class weight $w$ in the interval $[0.1, 5]$ and show that they are antagonistic, which demonstrates that weighted maxent can trade-off Precision and Recall.

Second, we show that our golden section search algorithm finds a good approximation of the optimum class-weight $w$, necessary for maximizing a specific $F_\beta(w)$, despite the violation of the unimodality of $F_\beta(w)$. We can identify the optimum weights by means of a *brute-force* approach, by which we try a large number of values for the weight of the target class (in practice, 50 values evenly distributed in $[0.1, 5]$). The *brute-force* is infeasible practical applications, because it requires training a large number of weighted maxent models. The comparison to the *brute-force* method is carried on the training set, because finding the appropriate class weight $w$ is part of model fitting, together with the estimation of the model weights $\lambda$.

Third, we demonstrate that the models that we fit are superior (i.e. yield better test $F_\beta$) than the maxent model. To this end, we compute $F_\beta$ for a range of values of $\beta \in [0, 1]$. We compare these results with the test $F_\beta$ that our algorithm delivers. For a reliable comparison, we also estimate the variance of the $F_\beta$ values – both for our method and for the baseline – by training on 20 bootstrap samples of the training set instead of the original train set.




(a)




(b)

Figure 1: Distribution of the samples in the space of features for the synthetic datasets: a) dataset A ; b) dataset B

## 9 Datasets

### 9.1 Synthetic datasets

We simulated two datasets, A and B, of 600 samples each of them with two equally populated classes and only two features. In dataset A the samples from class 0 are distributed as $\mathcal{N}(\mu_0^A, \Sigma_0^A)$, with $\mu_0^A = (2, 1)$ and $\Sigma_0^A = (1, 0.3)^\top I_2$. Class 1 is generated by $\mathcal{N}(\mu_1^A, \Sigma_1^A)$, with $\mu_1^A = (1, 2)$ and $\Sigma_1^A = (0.3, 1)^\top I_2$.

Dataset B consists of two symmetric spherical Gaussians - $\mathcal{N}(\mu_0^B, \Sigma_0^B)$ and $\mathcal{N}(\mu_1^B, \Sigma_1^B)$ with $\mu_0^B = (0.5, 1)$, $\Sigma_0^B = (0.3, 0.3)^\top I_2$, $\mu_1^B = (1, 0.8)$ and $\Sigma_1^B = (0.3, 0.3)^\top I_2$.

In Figure 1 we visualize both synthetic datasets. We used 400 of the samples for training and 200 for testing.

### 9.2 Twitter sentiment corpus

We used the Sanders Twitter Sentiment Corpus (http://www.sananalytics.com/lab/twitter-sentiment/), from which we filtered 3425 tweets, labeled as either *positive*, *negative* or *neutral*. We classified tweets that expressed a sentiment (either positive or negative), versus neutral tweets. The neutral tweets are about twice more than the positive and negative tweets together. For the experiments, we used $3081(90\%)$ tweets for training and $343$ $(10\%)$ for testing. We processed the tweets and obtained about $6095$ features. In order to avoid overfitting and speed up computations, we used a filter method based on Information Gain to remove uninformative features. We kept $60$ $(10\%)$ of the features for our experiments.

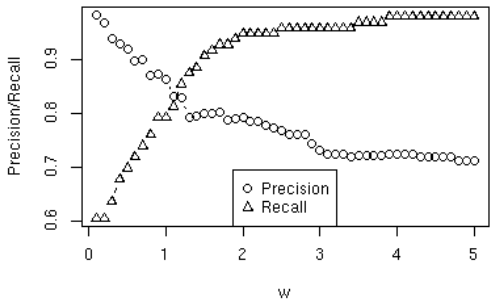
## 10 Experiments and results

By varying the weight of the target class, the weighted maximum entropy achieves Precision-Recall trade-off. Figure 2 clearly illustrates the trade-off, for the synthetic data A and the twitter sentiment data. Additionally, note that Precision and Recall are in equilibrium for a a weight that reflects the ratio of the class cardinalities, namely $w = 1$ for the balanced synthetic dataset A and $w = 2$, for the twitter corpus.

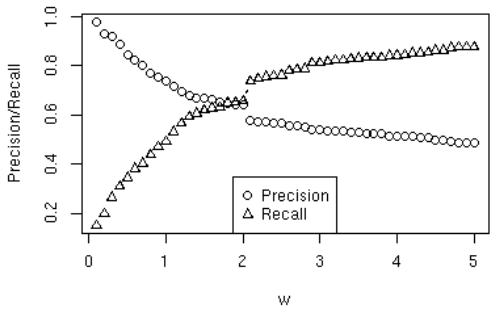
The *brute force* method reveals the shape of the $F_\beta(w)$, as a function of $\beta$ and $w$ (see Figure 4 a) and c)). Both of our datasets suggest that there is a critical value of $w$ which marks a switch point in the monotony of the $F_\beta(w)$ (regarded as a function of $\beta$). For $w$ smaller than the critical switch, $F_\beta(w)$ increases with $\beta$, and for $w$ larger than the switch, $F_\beta(w)$ decreases with $\beta$. This switch is probably directly related to the ratio of the class cardinalities and deserves further theoretical investigation.

Figures 4 a) and c) show also the 'path' that marks the maximum $F_\beta$ achievable for each $\beta$, in solid black line. The path corresponding to our golden search algorithm falls fairly close to that of the *brute force*, as shown by the dotted lines (marking the mean and one standard deviation to each side). Even if sometimes the optimal $w$ is not found exactly by the golden search, the $F_\beta$ is still very close to the optimum, as shown in Figures 4 b) and d). In fact, the optimum $F_\beta$ is always within one standard deviation from the expected value of our golden search algorithm.

Finally, we demonstrate that our method performs very well on the test set, compared to the simple maxent baseline. Figure 3 a) and b) show that the test $F_\beta$ is superior to the baseline, due to its ability to adapt the fitted model to the specific Precision - Recall trade-off, expressed by a value of $\beta$.




(a)



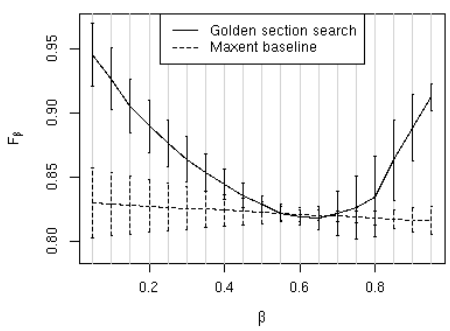

(b)

Figure 2: Precision-Recall trade-off on the train set by changing class-weights: a) synthetic dataset A; b) sentiment tweeter dataset.

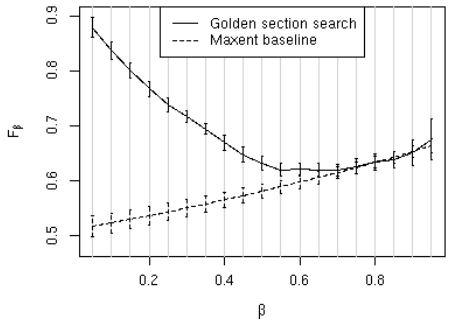
## 11 Limits and merits of the weighted maximum entropy

In this section we compare the weighted maximum entropy and the acceptance threshold method with the help of the two artificial data sets A and B shown on Figure 1. The acceptance threshold corresponds to a translation of the separating hyperplane obtained by the standard maximum entropy model. We show that acceptance threshold fails to fit the data well for most values of $\beta$, if the data resemble more dataset A than dataset B. In contrast, the weighted maxent is more adaptive, fitting nicely both datasets for all values of $\beta$.

It is rather clear that with translation we can achieve an optimal Precision/Recall trade-off for




(a)




(b)

Figure 3: Test $F_\beta$ for our method, compared to the maxent baseline. One standard deviation bars are added. a) synthetic data; b) twitter corpus.

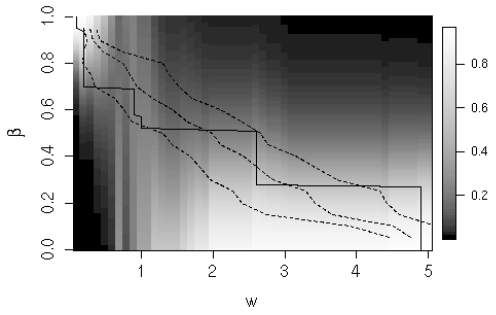
the synthetic data set B. Indeed, Figure 5 b) shows that the acceptance threshold and the weighted maximum entropy do result in virtually the same optimal $F_\beta$ values.

The optimal Precision/Recall trade-off for dataset A however requires additional rotation/tilting of the separating hyperplane that cannot be produced by adjusting the acceptance threshold. In line with this intuition Figure 5 a) demonstrates that the weighted likelihood settles at a better Precision-Recall pairs and consequently results in larger $F_\beta$ values.

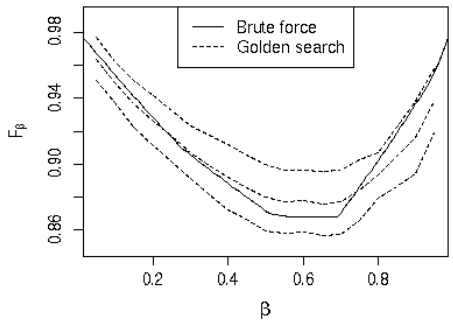
Clearly, in the general case the optimal shift of the separating plane is expected to have a rotation component that is unaccessible by simply adjusting the acceptance threshold.

## 12 Conclusion and future work

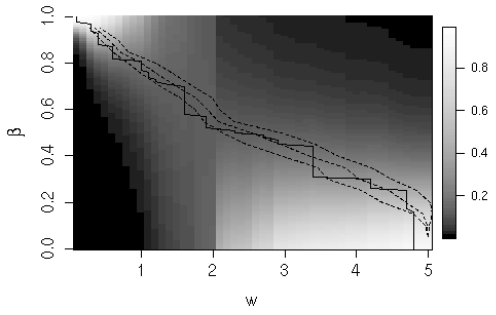
The main result of the paper is that the weighted maximum likelihood and the expected $F_\beta$ measure are simply two different ways to specify a particular trade-off between the objectives of the same multi-criteria optimization problem. Technically we unify these two approaches by viewing them as methods to pick a particular point from



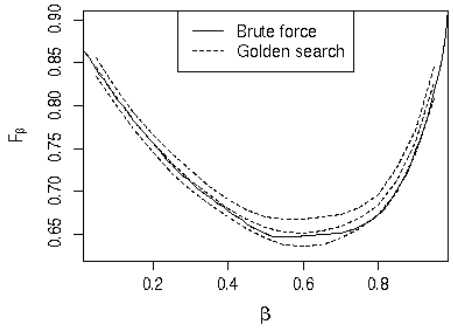

(a)




(b)

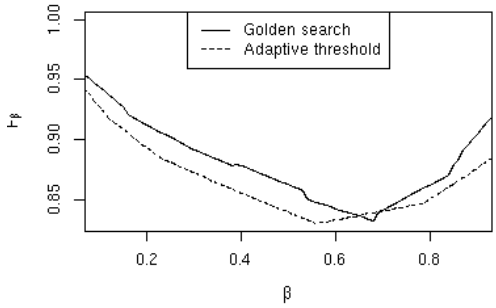



(c)

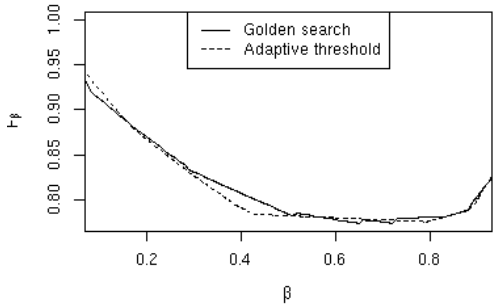



(d)

Figure 4: Heatmap showing in grayscale the $F_\beta(w)$ values obtained by the *brute force* method. The solid black line shows the optimal models for each beta. The dotted lines show the estimates given by the golden search: a) synthetic data; c) sentiment corpus. Comparison of the train $F_\beta$ obtained with the *brute force* (solid line) and with the golden section search (dotted line, with standard deviation): b) synthetic data; d) sentiment corpus.




(a)




(b)

Figure 5: Comparison of the acceptance threshold versus the weighted maximum likelihood on the stylized synthetic data: a) dataset A ; b) dataset B

the Pareto optimal set associated with a common multi-criteria optimization problem.

As a consequence each expected $F_\beta$ maximizer can be realized as a weighted maximum likelihood estimator and approximated via a class-wise weighted maximum likelihood estimator.

The presented results can be generalized to the regularized and multi-class case which is a subject for future work.

Furthermore, the proposed approach to view any probabilistic learning scheme as a specific trade-off between different objectives and thus to link it to the expected $F_\beta$ measure is general and can be applied beyond the maximum entropy framework.

The difficulty in exploiting the statement of Proposition 1 lies in the fact that it is not apriori clear how to choose the weights $w(\beta)$ for a given $\beta$. In a larger paper the authors will present algorithms maximizing the $\tilde{F}_\beta$ measure exploiting the theoretical results from this paper via adaptively finding the right weights. Even without a precise estimate for the weights the presented results give the qualitative connection between the Precision/Recall trade-off and the weights: if one aims at higher Precision then smaller weights are appropriate and conversely larger Recall is achieved via larger weights.

We showed with experiments on artificial and real data that using weighted maximum entropy we can achieve a desired Precision - Recall trade-off. We also presented an efficient algorithm based on golden section search, that approximates well the class weights at which the maximum $F_\beta$ is attained. We showed that on the test set, we achieve larger $F_\beta$ than the simple maximum entropy baseline.

## References

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

R. P. Brent. 1973. *Algorithms for Minimization Without Derivatives*. Prentice-Hall, Inc., Englewood Cliffs, New Jersery.

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

Matthias Ehrgott. 2005. *Multi Criteria Optimization*. Springer, Englewood Cliffs, New Jersery.

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

J. Kiefer. 1953. Sequential minimax search for a maximum. *Proceedings of the American Mathematical Society*, 4(3):502–506.

E. Minkov, R.C. Wang, A. Tomasic, and W.W. Cohen. 2006. NER systems that suit user's preferences: adjusting the recall-precision trade-off for entity extraction. In *Proceedings of NAACL*, pages 93–96.

Ye Nan, Kian Ming Adam Chai, Wee Sun Lee, and Hai Leong Chieu. 2012. Optimizing f-measure: A tale of two approaches. In *ICML*.

M. Simecková. 2005. Maximum weighted likelihood estimator in logistic regression.

D. L. Vandev and N. M. Neykov. 1998. About regression estimators with high breakdown point. *Statistics*, 32:111–129.