GRADIENT-BASED LEARNING FOR F-MEASURE AND OTHER PERFORMANCE METRICS

Anonymous authors
Paper under double-blind review

ABSTRACT

Many important classification performance metrics, e.g. F-measure, are non-differentiable and non-decomposable, and are thus unfriendly to gradient descent algorithm. Consequently, despite their popularity as evaluation metrics, these metrics are rarely optimized as training objectives in neural network community. In this paper, we propose an empirical utility maximization scheme with provable learning guarantees to address the non-differentiability of these metrics. We then derive a strongly consistent gradient estimator to handle non-decomposability. These innovations enable end-to-end optimization of these metrics with the same computational complexity as optimizing a decomposable and differentiable metric, e.g. cross-entropy loss.

1 INTRODUCTION

Different classification performance metrics measures different aspects of a classifier's behavior. For example, F-measure (Van Rijsbergen (1974)), compared to metrics such as accuracy, is better at evaluating a classifier's performance when it encounters a sample belonging to a class that occurs with low frequency. Ideally, we can acquire a classifier with very tailored behavior by optimizing the classifier with respect to a carefully chosen performance metric. Unfortunately, many performance metrics are non-differentiable and non-decomposable, which renders it very difficult to optimize neural network classifiers with these metrics as training objective.

In this paper, we propose a method that enables gradient-based learning for these performance metrics. Our contributions are the following:

- We propose a learning algorithm based on empirical utility maximization for a class of performance metrics and prove its generalization and consistency.
- We propose a strongly consistent gradient estimator that enables efficient gradient-based maximization of empirical utility.
- We demonstrate experimentally that the binary F_1 score of neural network classifiers can be optimized on datasets of decent scale.

We organize this paper as follows. In Section 2, we sketch our approach for binary F_1 score to provide an overview. In Section 3, we present our method in its general form. We review related work in Section 4 and provide experimental results in Section 5.

2 GRADIENT-BASED LEARNING FOR BINARY F1 SCORE

2.1 PROBABILISTIC CLASSIFIER

Given a feature vector $\boldsymbol{x} \in \mathcal{X} \subset \mathbb{R}^N$, a probabilistic classifier h first infers a posterior $p(\cdot|\boldsymbol{x})$ over a discrete output space $\mathcal{Y}$ and then samples an output from the posterior, i.e. $h(\boldsymbol{x}) \sim p(\cdot|\boldsymbol{x})$. In practice, $p(\cdot|\boldsymbol{x})$ is typically the output of a neural network with softmax layer on its top. When the posterior is parameterized, e.g. being implemented as a neural network, we denote it as $p_\theta(\cdot|\boldsymbol{x})$ and the corresponding probabilistic classifier as h_theta.

Given a posterior $p(\cdot|\boldsymbol{x})$, a deterministic classifier results from the inference rule $h(\boldsymbol{x}) = \text{argmax}_{y\in\mathcal{Y}} p(y|\boldsymbol{x})$.The difference between probabilistic and deterministic inference rules is negligible when the posterior is very concentrated. Although deterministic classifiers are more popular in the literature, in this paper we only consider probabilistic classifiers and leave it as future work to investigate the case where a probabilistic classifier is replaced by a deterministic one.

2.2 F-MEASURE

Consider the case of binary classification, where $\mathcal{Y} = \{0, 1\}$ with 1 and 0 corresponding to the positive and negative class. Given an i.i.d. dataset $D = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$ consisting of n pairs of feature vector and ground truth, let $\hat{y}_i$ denote the label predicted by a classifier h given $\boldsymbol{x}_i$ (not necessarily deterministically). Let $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)$ and $\boldsymbol{y} = (y_1, ..., y_n)$. Then the true-positive, false-positive, false-negative and true-negative rate corresponding to $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ are defined as

$$\text{tp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big(\hat{y}_i = 1 \wedge y_i = 1\big) \qquad \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big(\hat{y}_i = 1 \wedge y_i = 0\big)$$

$$\text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big(\hat{y}_i = 0 \wedge y_i = 1\big) \qquad \text{tn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}\big(\hat{y}_i = 0 \wedge y_i = 0\big)$$

where $\mathbb{I}$ denotes indicator function. The precision and recall are defined as

$$\text{precision}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\text{tp}(\hat{\boldsymbol{y}}, \boldsymbol{y})}{\text{tp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y})} \qquad \text{recall}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{\text{tp}(\hat{\boldsymbol{y}}, \boldsymbol{y})}{p_D^+} \tag{1}$$

where $p_D^+ := \frac{1}{n}\sum_{i=1}^{n} \mathbb{I}(y_i = 1)$ denotes the proportion of samples in D that belong to positive class. The binary F-measure is defined as (Van Rijsbergen (1974)):

$$F_\beta(\hat{\boldsymbol{y}}, \boldsymbol{y}) = (1 + \beta^2) \cdot \frac{\text{precision}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \cdot \text{recall}(\hat{\boldsymbol{y}}, \boldsymbol{y})}{\beta \cdot \text{precision}(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \text{recall}(\hat{\boldsymbol{y}}, \boldsymbol{y})} \qquad \beta > 0 \tag{2}$$

or equivalently,

$$F_\beta(\hat{\boldsymbol{y}}, \boldsymbol{y}) = (1 + \beta^2) \cdot \frac{p_D^+ - \text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y})}{(1 + \beta^2)p_D^+ - \text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y})} \qquad \beta > 0 \tag{3}$$

which is more convenient for our purpose.

We will refer to $F_\beta(\hat{\boldsymbol{y}}, \boldsymbol{y})$ as *data-dependent* binary F_β-measure because it is evaluated on a specific set of data with pairs of ground truth and prediction vectors. F_β is non-differentiable because it is a composition of indicator functions. Nor does it decompose over samples in D. More precisely, people are not aware of any function f_β that only depends on per sample ground-truth and prediction such that

$$F_\beta(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \frac{1}{n}\sum_{i=1}^{n} f_\beta(\hat{y}_i, y_i)$$

In the following we propose an empirical utility maximization scheme for optimizing the F_β-measure of probabilistic classifiers. For ease of exposition, in this section we focus on binary F_1-measure, a.k.a. binary F_1 score. In Section-3, we will extend the method presented in this section to a family of non-decomposable and non-differentiable performance metrics, including F_β-measure for multi-class classification.

2.3 GRADIENT-BASED LEARNING FOR BINARY F1 SCORE

We consider a parameterized binary probabilistic classifier h_θ. By linearity of expectation and i.i.d. assumption,

$$\mathbb{E}_{\hat{\boldsymbol{y}}, \boldsymbol{y}}[\text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y})] = \mathbb{P}(\hat{\text{y}} = 1 \wedge \text{y} = 0)$$

Similarly, $\mathbb{E}_{\hat{\boldsymbol{y}}, \boldsymbol{y}}[\text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y})] = \mathbb{P}(\bar{\text{y}} = 0 \wedge \text{y} = 1)$. Let $\overline{\text{fn}}(h_\theta) := \mathbb{E}_{\hat{\boldsymbol{y}}, \boldsymbol{y}}[\text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y})]$ and $\overline{\text{fp}}(h_\theta) := \mathbb{E}_{\hat{\boldsymbol{y}}, \boldsymbol{y}}[\text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y})]$. It follows from law of large number that

$$\lim_{n\to\infty} \text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \overline{\text{fn}}(h_\theta) \qquad \lim_{n\to\infty} \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \overline{\text{fp}}(h_\theta)$$

with probability 1. Thus on sufficiently large datasets,

$$\text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \approx \overline{\text{fn}}(h_\theta) \qquad \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \approx \overline{\text{fp}}(h_\theta)$$

With these approximate identities, we have

$$F_1(\hat{\boldsymbol{y}}, \boldsymbol{y}) = 2 \cdot \frac{p_D^+ - \text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y})}{2p_D^+ - \text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) + \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y})} \approx 2 \cdot \frac{p_D^+ - \overline{\text{fn}}(h_\theta)}{2p_D^+ - \overline{\text{fn}}(h_\theta) + \overline{\text{fp}}(h_\theta)} := \bar{F}_1(h_\theta)$$

which implies that the F_1 score of *any* predictions of h_θ on *any* sufficiently large dataset is close to $\bar{F}_1(h_\theta)$. We call $\bar{F}_1(h_\theta)$ *expected utility* and state the precise meaning of $\bar{F}_1(\hat{\boldsymbol{y}}, \boldsymbol{y}) \approx F_1(h_\theta)$ in Section-3. The key point is that we can optimize $\bar{F}_1(h_\theta)$ instead of $F_1(\hat{\boldsymbol{y}}, \boldsymbol{y})$ if we are interested in the F_1 score of h_θ on large datasets. However, $\overline{\text{fn}}(h_\theta)$ and $\overline{\text{fp}}(h_\theta)$ are unknown because they are expectations taken over data distribution (and classifier's posterior). Consequently, we have to estimate them by sampling from data distribution in order to compute $\bar{F}_1(h_\theta)$.

Let $p^+ = \mathbb{P}(y = 1)$ denote the probability that a positive sample occurs, which can be estimated by the frequency of positive samples in training set. Let n^+ denote the number of positive samples in training set. Assume that the data distribution admits a density function (i.e. the data distribution is absolutely continuous w.r.t. Lebesgue measure), and denote the density function by p. We have

$$\begin{aligned}
\overline{\text{fn}}(h_\theta) &= \mathbb{P}\big(\bar{\text{y}} = 0 \wedge \text{y} = 1\big) \\
&= \int_{\mathcal{X}} \mathbb{P}(h_\theta(\boldsymbol{x}) = 0) p(\boldsymbol{x}, 1) \mathrm{d}\boldsymbol{x} \\
&= \int_{\mathcal{X}} p_\theta(0|\boldsymbol{x}) p(\boldsymbol{x}|1) p^+ \mathrm{d}\boldsymbol{x} \\
&= p^+ \int_{\mathcal{X}} p_\theta(0|\boldsymbol{x}) p(\boldsymbol{x}|1) \mathrm{d}\boldsymbol{x} \\
&= p^+ \mathbb{E}_{\boldsymbol{x} \sim p(\cdot|1)}[p_\theta(0|\boldsymbol{x})] \\
&\approx \frac{p^+}{n^+} \sum_{i=1}^{n^+} p_\theta(0|\boldsymbol{x}_i^+) := \widehat{\text{fn}}_D(h_\theta)
\end{aligned} \tag{4}$$

where $\boldsymbol{x}_1^+$, ..., $\boldsymbol{x}_{n^+}^+$ are feature vectors of samples belonging to the positive class in training set D. Similarly, we have

$$\overline{\text{fp}} = (h_\theta)\mathbb{P}\big(\bar{\text{y}} = 1 \wedge \text{y} = 0\big) \approx \frac{p^-}{n} \sum_{i=1}^{n} p_\theta(1|\boldsymbol{x}_i^-) := \widehat{\text{fp}}_D(h_\theta)$$

where $p^- := \mathbb{P}(\text{y} = 0)$, $n^- := \sum_{(\boldsymbol{x}, y) \in D} \mathbb{I}(y = 0)$, and $\boldsymbol{x}_1^-$, ..., $\boldsymbol{x}_n^-$ are feature vectors of samples in D belonging to the negative class. Thus $\bar{F}_1(h_\theta)$ can be estimated as following:

$$\bar{F}_1(h_\theta) = 2 \cdot \frac{p^+ - \overline{\text{fn}}(h_\theta)}{2p^+ - \overline{\text{fn}}(h_\theta) + \overline{\text{fp}}(h_\theta)} \approx 2 \cdot \frac{p^+ - \widehat{\text{fn}}_D(h_\theta)}{2p^+ - \widehat{\text{fn}}_D(h_\theta) + \widehat{\text{fp}}_D(h_\theta)} := \hat{F}_D(h_\theta) \tag{5}$$

Interestingly, although $\text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ and $\text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y})$ are not differentiable, the estimators of their expectations, $\widehat{\text{fn}}_D(h_\theta)$ and $\widehat{\text{fp}}_D(h_\theta)$, are differentiable w.r.t. θ if p_θ is differentiable. Because $\hat{F}_D$ is differentiable w.r.t. $\widehat{\text{fn}}(h_\theta)$ and $\widehat{\text{fp}}(h_\theta)$, $\nabla_\theta \bar{F}_1(h_\theta)$ can be computed by chain rule. Consequently, gradient descent can be applied to optimize $\hat{F}_1$.

We will refer $\hat{F}_D(\theta)$ as the *empirical utility* of the expected utility $\bar{F}_1(h_\theta)$. They correspond to empirical and expected risk in the classical empirical risk minimization principle of statistical learning theory (Vapnik (1992)). We use the term "empirical utility maximization" because we would like to maximize, instead of minimize these performance metrics. There are two fundamental questions for every empirical risk minimization style learning algorithm, as the number of samples increases:

- Generalization. Given h_θ, does $\hat{F}_D(h_\theta)$ converge to $\bar{F}_1(h_\theta)$?
- Consistency. Does $\text{argmax}_\theta \hat{F}_D(h_\theta)$ converge to $\text{argmax}_\theta \bar{F}_1(h_\theta)$?

We will address these two questions in Section-3. For the moment let us consider a practical issue: how to maximize empirical utility $\bar{F}_D(h_\theta)$ efficiently with gradient descent?

2.4 GRADIENT ESTIMATOR

Large training sets are necessary for the approximation Eq (5) to be accurate. Consequently, in order to optimize $\hat{F}_D(h_\theta)$ efficiently via minibatch gradient descent, $\nabla_\theta \hat{F}_D(h_\theta)$ has to be estimated by $\hat{F}_B(h_\theta)$, where $B \subset D$ is a mini-batch. However, mini-batch gradient descent requires that

$$\mathbb{E}_B\left[\nabla_\theta \hat{F}_B(h_\theta)\right] = \nabla_\theta \hat{F}_D(h_\theta) \tag{6}$$

Suppose $\hat{F}_D(h_\theta)$ is decomposable, i.e.

$$\exists \hat{f}, \hat{F}_D(h_\theta) = \frac{1}{|D|} \sum_{(\boldsymbol{x},y)\in D} \hat{f}(h_\theta(\boldsymbol{x}), y)$$

then Eq (6) simply follows from the linearity of differentiation and expectation. However, as $\hat{F}_D(h_\theta)$ is non-decomposable, it becomes unlikely that $\nabla_\theta \hat{F}_B(h_\theta)$ is an unbiased estimator of $\nabla_\theta \hat{F}_D(h_\theta)$. Fortunately, $\nabla_\theta \hat{F}_B(h_\theta)$ is a strongly consistent estimator of $\nabla_\theta \hat{F}_D(h_\theta)$. More precisely, assuming that D contains infinitely many samples, we have

$$\mathbb{P}\left(\lim_{|B|\to\infty} \nabla_\theta \hat{F}_B(h_\theta) = \nabla_\theta \hat{F}_D(h_\theta)\right) = 1 \tag{7}$$

More interestingly, the noise incurred by estimating $\nabla_\theta \hat{F}_D(h_\theta)$ with $\nabla_\theta \hat{F}_B(h_\theta)$ can be further controlled. In the following we omit the dependence on h_θ for brevity. Let $\boldsymbol{\phi}_D = \left(\widehat{fn}_D(h_\theta), \widehat{fp}_D(h_\theta)\right)$ and $\boldsymbol{\phi}_B = \left(\widehat{fn}_B(h_\theta), \widehat{fp}_B(h_\theta)\right)$. Let $\widehat{\boldsymbol{J}}_D$ and $\widehat{\boldsymbol{J}}_B$ denotes the Jacobian of $\widehat{\boldsymbol{\phi}}$ and $\widehat{\boldsymbol{\phi}}_B$ w.r.t. θ. Let $||\cdot||$ denote matrix norm and $|\cdot|$ denote vector norm. By chain rule,

$$\begin{aligned}
\nabla_\theta \hat{F}_B &= \nabla_{\boldsymbol{\phi}_B} \hat{F}_B \cdot \widehat{\boldsymbol{J}}_B \\
&= \left(\nabla_{\boldsymbol{\phi}_D} \hat{F}(h_\theta) + \boldsymbol{\epsilon}\right) \cdot \left(\widehat{\boldsymbol{J}} + \boldsymbol{\mathcal{E}}\right) \\
&= \nabla_{\boldsymbol{\phi}_D} \hat{F} \cdot \widehat{\boldsymbol{J}} + \nabla_{\boldsymbol{\phi}_D} \hat{F} \cdot \boldsymbol{\mathcal{E}} + \boldsymbol{\epsilon} \cdot \widehat{\boldsymbol{J}} + \boldsymbol{\epsilon} \cdot \boldsymbol{\mathcal{E}}
\end{aligned} \tag{8}$$

where $\nabla_{\boldsymbol{\phi}} \hat{F}(h_\theta) \cdot \widehat{\boldsymbol{J}}(h_\theta)$ is true gradient and $\boldsymbol{\epsilon} \cdot \boldsymbol{\mathcal{E}}$ is negligible. The error matrix $\boldsymbol{\mathcal{E}} = \widehat{\boldsymbol{J}}_B(h_\theta) - \widehat{\boldsymbol{J}}_D(h_\theta)$ is intrinsic in the sense that it results immediately from estimating $\nabla_\theta \widehat{fn}_D(h_\theta)$ and $\nabla_\theta \widehat{fp}_D(h_\theta)$ with $\nabla_\theta \widehat{fn}_B(h_\theta)$ and $\nabla_\theta \widehat{fp}_B(h_\theta)$. However, we can control the error $\boldsymbol{\epsilon} \cdot \widehat{\boldsymbol{J}}$ because $|\boldsymbol{\epsilon} \cdot \widehat{\boldsymbol{J}}| \leq ||\widehat{\boldsymbol{J}}|| \cdot |\boldsymbol{\epsilon}|$ and we can control $||\widehat{\boldsymbol{J}}||$ by limiting $|\theta|$ and the norm of intermediate activation when p_θ is a neural network (He et al. (2015)). Despite these technicalities, the trick is very easy to implement: batch-norm (Ioffe & Szegedy (2015)) and weight decay will suffice. The resultant algorithm is the following:

Algorithm 1 Gradient-based learning for binary F_1 score

Require: classifier h_θ, dataset D, batch size b, learning rate α, weight decay factor λ
  $p^+ \leftarrow \frac{1}{|D|} \sum_{(\boldsymbol{x},y)\in D} \mathbb{I}(y=1)$
  $p^- \leftarrow \frac{1}{|D|} \sum_{(\boldsymbol{x},y)\in D} \mathbb{I}(y=0)$
  while terminating criterion not satisfied **do**
    Sample $B_+ = \{(\boldsymbol{x}_1^+, 1), ..., (\boldsymbol{x}_b^+, 1)\}$ from D
    Sample $B_- = \{(\boldsymbol{x}_1^-, 0), ..., (\boldsymbol{x}_b^-, 0)\}$ from D
    fn $\leftarrow \frac{p^+}{b} \sum_{i=1}^{b} p_\theta(0|\boldsymbol{x}_1^+)$
    fp $\leftarrow \frac{p^-}{b} \sum_{i=1}^{b} p_\theta(1|\boldsymbol{x}_1^-)$
    $\delta \leftarrow \nabla_\theta\big(F_1(\text{fn}, \text{fp}) - \lambda \cdot ||\theta||_2\big)$
    $\theta \leftarrow \theta + \alpha \cdot \delta$
  end while

3 GRADIENT-BASED LEARNING FOR A CLASS OF PERFORMANCE METRICS

Binary F-measure in fact belong to a class of metrics that are well behaved functions of the confusion matrix. In this section, we propose a gradient-based learning algorithm that extends the approach illustrated in previous section to this class of metrics. We state two theorems concerning the generalization and consistency of the proposed algorithm. We defer the proofs of these theorems to appendix. We begin with a formal specification of this class of metrics, which relies on the definition of confusion matrix:

Definition 1. *Given a dataset $D = \{(\boldsymbol{x}_1, y_1), ..., (\boldsymbol{x}_n, y_n)\}$, let $\boldsymbol{y} = (y_1, ..., y_n)$ denote the vector of ground truth and $\hat{\boldsymbol{y}} = (\hat{y}_1, ..., \hat{y}_n)$ denote the vector of predictions. Then the corresponding data-dependent confusion matrix $C(\hat{\boldsymbol{y}}, \boldsymbol{y})$ is defined as*

$$\Big(C(\hat{\boldsymbol{y}}, \boldsymbol{y})\Big)_{ij} = \frac{1}{n}\sum_{k=1}^{n} \mathbb{I}\big(\hat{\boldsymbol{y}}_k = i \wedge y_k = j\big) \qquad 0 \leq i, j \leq |\mathcal{Y}| - 1$$

Given a probability measure $\mathbb{P}$ over $\mathcal{X} \times \mathcal{Y} \times \mathcal{Y}$, i.e. triples of feature vector, ground-truth and prediction, the expected confusion matrix $\bar{C}$ of a probabilistic classifier h is a $|\mathcal{Y}| \times |\mathcal{Y}|$ matrix defined as

$$\Big(\bar{C}(h_\theta)\Big)_{ij} = \mathbb{P}(h(\boldsymbol{x}) = i \wedge y = j) \qquad 0 \leq i, j \leq |\mathcal{Y}| - 1$$

The confusion matrix is well-defined for both single-label and multi-label classification (although these two scenarios impose different constraints on its entries). In the case of binary classification,

$$\Big(C(\hat{\boldsymbol{y}}, \boldsymbol{y})\Big)_{ij} = \begin{bmatrix} \text{tn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) & \text{fn}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \\ \text{fp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) & \text{tp}(\hat{\boldsymbol{y}}, \boldsymbol{y}) \end{bmatrix} \qquad \Big(\bar{C}(h_\theta)\Big)_{ij} = \begin{bmatrix} \overline{\text{tn}}(h_\theta) & \overline{\text{fn}}(h_\theta) \\ \overline{\text{fp}}(h_\theta) & \overline{\text{tp}}(h_\theta) \end{bmatrix}$$

As in previous section, given a training set D, we have the estimator

$$\Big(\hat{C}_D(h_\theta)\Big)_{ij} = \frac{p_j}{n_j}\sum_{k=1}^{n_j} p_\theta(i|\boldsymbol{x}_k^j)$$

where p_j is the frequency of the j-th class, n_j is the number of samples in D that belong to the j-th class, and $\boldsymbol{x}_1^j, ..., \boldsymbol{x}_{n_j}^j$ are feature vectors of samples that belong to the j-th class. Entry-wise convergence in probability follows from law of large number

$$\mathbb{P}\Big(\lim_{|D|\to\infty} C(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \bar{C}(h_\theta)\Big) = 1 \qquad \mathbb{P}\Big(\lim_{|D|\to\infty} \hat{C}_D(h_\theta) = \bar{C}(h_\theta)\Big) = 1 \tag{9}$$

Many performance metrics are functions of the confusion matrix. For example, accuracy$(h_\theta) = \sum_{i=1}^{|\mathcal{Y}|} \bar{C}_{ii}(h_\theta)$. F_β measure for multi-class classification can be defined in term of entries of the confusion matrix as following. The data-dependent per class frequency, false positive and false negative, defined as functions of $C(\hat{\boldsymbol{y}}, \boldsymbol{y})$, are respectively

$$p_i = \sum_{j=1}^{|\mathcal{Y}|} C_{ji} \qquad \text{fp}_i = \sum_{j\neq i} C_{ij} \qquad \text{fn}_i = \sum_{j\neq i} C_{ji} \qquad i = 1, ..., |\mathcal{Y}|$$

where we omit the dependence on $\hat{\boldsymbol{y}}$ and $\boldsymbol{y}$ for brevity. The data-dependent macro F-measure is defined in term of these quantities as

$$F_\beta = \frac{1+\beta^2}{|\mathcal{Y}|}\sum_{i=1}^{|\mathcal{Y}|} \frac{p_i - \text{fn}_i}{(1+\beta^2)p_i - \text{fn}_i + \text{fp}_i} \qquad \beta > 0$$

and the data-dependent micro F-measure is defined in term of these quantities as

$$F_\beta = (1+\beta^2) \cdot \frac{\sum_{i=1}^{|\mathcal{Y}|} p_i - \sum_{i=1}^{|\mathcal{Y}|} \text{fn}_i}{(1+\beta^2)\sum_{i=1}^{|\mathcal{Y}|} p_i - \sum_{i=1}^{|\mathcal{Y}|} \text{fn}_i + \sum_{i=1}^{|\mathcal{Y}|} \text{fp}_i} \qquad \beta > 0$$

Replacing $C(\hat{\boldsymbol{y}}, \boldsymbol{y})$ by $\bar{C}(h_\theta)$ and $\hat{C}(h_\theta)$ respectively results in expected and empirical F-measure.

We now formally define the class of metrics that we are interested in, namely the class of well-behaved metrics. In the following, we will consider C, $C(\hat{\boldsymbol{y}}, \boldsymbol{y})$ and $\bar{C}_E$ as vectors of dimension $|\mathcal{Y}| \times |\mathcal{Y}|$ instead of matrices.

Definition 2. *We say that a performance metric $F : K \mapsto \mathbb{R}$, where K is a compact subset of $\mathbb{R}^{|\mathcal{Y}|\times|\mathcal{Y}|}$, is well-behaved if F is differentiable and its gradient is continuous in the interior of K.*

Please refer to appendix for a non-exhaustive list of well-behaved performance metrics. Given a well behaved performance metric F, its corresponding data-dependent, expected and empirical utility are respectively defined as $F(C(\hat{\boldsymbol{y}}, \boldsymbol{y}))$, $F(\bar{C}(h_\theta))$ and $F(\hat{C}(h_\theta))$. The following theorem establishes asymptotic equivalence between these three kinds of utilities.

Theorem 1. *If F is a well-behaved performance metric and C_n is a strongly consistent estimator of C, i.e.*

$$\mathbb{P}\Big(\lim_{n\to\infty} F\big(C_n\big) = F(C)\Big) = 1$$

where convergence in probability is entry-wise, then $F\big(C_n\big)$ is a strongly consistent estimator for $F(C)$, i.e.

$$\mathbb{P}\Big(\lim_{|D|\to\infty} F\big(\bar{C}_D\big) = F(C)\Big) = 1$$

As a consequence of this theorem, it follows from Eq (9) that

$$\mathbb{P}\Big(\lim_{|D|\to\infty} F\big(C(\hat{\boldsymbol{y}}, \boldsymbol{y})\big) = F\big(\bar{C}(h_\theta)\big)\Big) = 1 \qquad \mathbb{P}\Big(\lim_{|D|\to\infty} F\big(\hat{C}_D(h_\theta)\big) = F\big(\bar{C}(h_\theta)\big)\Big) = 1$$

i.e. both $F\big(C(\hat{\boldsymbol{y}}, \boldsymbol{y})\big)$ and $F\big(\bar{C}(h_\theta)\big)$ are asymptotically equivalent to $F\big(\bar{C}(h_\theta)\big)$. Thus decent empirical utility $F\big(\bar{C}(h_\theta)\big)$ implies decent data-dependent utility $F\big(C(\hat{\boldsymbol{y}}, \boldsymbol{y})\big)$ on any sufficiently large dataset. As a special case,

$$\mathbb{P}\Big(\lim_{|D|\to\infty} F_1(\hat{\boldsymbol{y}}, \boldsymbol{y}) = \bar{F}_1(h_\theta)\Big) = 1 \qquad \mathbb{P}\Big(\lim_{|D|\to\infty} \hat{F}_1(h_\theta) = \bar{F}_1(h_\theta)\Big) = 1$$

which justifies Eq (2.3) and Eq (5) when the dataset of interest is sufficiently large.

Next we consider the issue of gradient estimation in this general setting.

Theorem 2. *For a well behaved performance metric F, $\nabla_\theta F\big(\hat{C}_B(h_\theta)\big)$ is a strongly consistent estimator of $\nabla_\theta F\big(\bar{C}(h_\theta)\big)$, where $B \subset D$ is a mini-batch. More precisely, assuming that D contains infinitely many samples, we have*

$$\mathbb{P}\Big(\lim_{|B|\to\infty} \nabla_\theta F\big(\hat{C}_B(h_\theta)\big) = \nabla_\theta F\big(\hat{C}_D(h_\theta)\big)\Big) = 1$$

As proved in Chen & Luss (2018), many convergence guarantees for mini-batch gradient descent holds with probability 1 for strongly consistent gradient estimator. As illustrated in Eq (8), batch-norm (Ioffe & Szegedy (2015)) and weight decay can help control the noise of estimator. Please refer to Algorithm-3 for full algorithm.

Finally, we state two theorems concerning the generalization and consistency of this algorithm. Rate of converges are omitted for brevity.

Theorem 3. *(Generalization) For a well behaved performance metric F, for all $\epsilon > 0$,*

$$\lim_{|D|\to\infty} \mathbb{P}\big(\big|F(\hat{C}_D(h_\theta)) - F(\bar{C}(h_\theta))\big| < \epsilon\big) = 1$$

Theorem 4. *(Consistency) For a well behaved performance metric F, with appropriate constraints on capacity of the parametric model p_θ (see proof for details), for all $\epsilon > 0$,*

$$\lim_{|D|\to\infty} \mathbb{P}\big(\big|\arg\max_\theta F(\hat{C}_D(h_\theta)) - \arg\max_\theta F(\bar{C}(h_\theta))\big| < \epsilon\big) = 1$$

Algorithm 2 Gradient-based learning for well behaved performance metrics

Require: classifier h_θ, well-behaved metric F, dataset D, batch size b, learning rate α
  for $i = 1, ..., |\mathcal{Y}|$ **do**
    $p_i \leftarrow \frac{1}{|D|} \sum_{(\boldsymbol{x},y)\in D} \mathbb{I}(y = i)$
  end for
  while terminating criterion not satisfied **do**
    for $i = 1, ..., |\mathcal{Y}|$ **do**
      Sample $B_i = \{(\boldsymbol{x}_1^i, i), ..., (\boldsymbol{x}_b^i, i)\}$ from D
      Compute $p_\theta(\cdot|\boldsymbol{x}_1^i)$, ..., $p_\theta(\cdot|\boldsymbol{x}_b^i)$
    end for
    for $i = 1, ..., |\mathcal{Y}|$ **do**
      for $j = 1, ..., |\mathcal{Y}|$ **do**
        $C_{ij} \leftarrow \frac{p}{b} \sum_{k=1}^{b} p_\theta(i|x_k^j)$
      end for
    end for
    $\delta \leftarrow \nabla_\theta\big(F(C) - ||\theta||_2\big)$
    $\theta \leftarrow \theta + \alpha \cdot \delta$
  end while

4 RELATED WORK

Optimization of non-decomposable and non-differentiable performance metrics, especially F-measure, has been extensively studied. The heuristic algorithm considered in Jansche (2005) and Pastor-Pellicer et al. (2013) is essentially Algorithm-(2.4) without techniques that stabilize gradient estimation. However, Jansche (2005) and Pastor-Pellicer et al. (2013) are not very well motivated theoretically and provide little mathematical insight into the algorithm. Also, as shown in Section-5, applying this heuristic algorithm without stabilization techniques can easily result in non-convergent models, even for a three-layer fully-connected network.

Another series of papers (Joachims (2005), Kar et al. (2014) and Narasimhan et al. (2015)) study optimizing differentiable lower bounds of various non-decomposable and non-differentiable binary classification metrics for linear classifiers. Despite proved learning guarantees for linear classifier, as reported in Sanyal et al. (2018), these lower bound methods are not very promising when applied to neural networks.

Thresholding is a computationally economic method if we only consider binary classification. Koyejo et al. (2014) proves that the optimal classifier with respect to a family of binary classification metrics, including F-measure, is appropriately thresholded Bayes classifier. Given an approximation of Bayes classifier, we can approach the optimal threshold via grid search. However, it remains unknown how to generalize thresholding to multi-class classification. More importantly, for binary classification, when training set is extremely imbalanced, it can be very difficult to train a classifier that approximates Bayes classifier very well.

The computational cost of aforementioned methods roughly equals that of training classifiers with standard classification losses such as cross-entropy. As proved in Parambath et al. (2014) and Koyejo et al. (2014), optimization of many performance metrics, including F-measure, can be reduced to weighted classification. Unfortunately, the optimal weight is in general unknown and has to be approached via an expensive grid search (see Section-5). Despite its computational cost, unlike thresholding, this method can perform very well even when training set is extremely balanced because of weighting. Eban et al. (2016) proposes a similar method that performs well for neural networks coupled with the AUCPR metric.

Regarding theory, the equivalence between data-dependent and expected utility of F-measure was first proved in Nan et al. (2012) and then generalized in Dembczyński et al. (2017) to p-Lipschitz binary classification performance metrics.

Table 1: Dataset statistics and results; GS and EUM refer to grid-search and expected utility method, respectively

DATASET	**# FEATS**	**# SAMPS**	**% POS**	F_1 **GS**	F_1 **EUM**
Adult	108	48,842	23.93	**0.701**	0.689
CIFAR10	3072	60,000	10.00	0.630	**0.635**
Letter	16	20,000	3.92	**0.990**	0.975
Covtype	54	581,012	1.63	0.691	**0.725**
CIFAR100	3072	60,000	1.00	0.350	**0.392**
KDDCup08	117	102,294	0.61	0.543	**0.556**

5 EXPERIMENTS

We evaluate our method on the following datasets: Letter[1], Adult[2], Covertype[3], KDDCup08[4], CIFAR10 and CIFAR100 (Krizhevsky (2009)). The performance metric to optimize is binary F_1 score. We use a three-layer fully-connected network in our experiments, with batch-norm enabled. The statistics of these datasets are summarized in Table-5 (number of features, number of samples, percentage of positive samples). For multi-class datasets (Letter, Covertype, CIFAR10 and CIFAR100), we designate one class as the positive class and leave the rest as the negative class. We compare Algorithm-(2.4) with the following baseline (Parambath et al. (2014) and Koyejo et al. (2014)):

$$\theta^* \leftarrow \underset{\theta,\lambda\in(0,1)}{\arg\max} \frac{1}{|D|} \sum_{(\boldsymbol{x},y)\in D} l(h_\theta(\boldsymbol{x}), y)\big(\lambda\mathbb{I}(y=1) + (1-\lambda)\mathbb{I}(y=1)\big)$$

where l denotes cross-entropy loss. We let $\lambda = 0.1, 0.2, 0.3, ..., 0.9$, and apply gradient descent to optimize θ given a fixed λ. As proved in Parambath et al. (2014) and Koyejo et al. (2014), this baseline should yield approximately optimal solution (at cost considerably higher than Algorithm-(2.4) because we have to optimize θ for every λ). In our case, the baseline is 8 times slower than Algorithm-(2.4). We apply weight decay to both methods and find that in general weight decay improves the performance of Algorithm-(2.4) while damages the performance of baseline. For Covertype dataset, Algorithm-2.4 cannot converge without weight decay. We report results in Table-5, where "F_1 GS" refers to the F_1 score of baseline and "F_1 EUM" refers to the F_1 score of Algorithm-2.4.

6 CONCLUSION

We propose an empirical utility maximization scheme that enables efficient gradient-based learning for a class of non-decomposable and non-differentiable classification performance metrics. We inquire into the proposed scheme mathematically and present preliminary experiments that validate our approach. We leave it as future work to experiment on deeper neural networks, larger datasets, and more complex performance metrics.

[1]https://archive.ics.uci.edu/ml/datasets/letter+recognition
[2]https://archive.ics.uci.edu/ml/datasets/adult
[3]https://archive.ics.uci.edu/ml/datasets/covertype
[4]http://www.kdd.org/kdd-cup/view/kdd-cup-2008/

REFERENCES

Jie Chen and Ronny Luss. Stochastic gradient descent with biased but consistent gradient estimators. *CoRR*, abs/1807.11880, 2018. URL `http://arxiv.org/abs/1807.11880`.

Krzysztof Dembczyński, Wojciech Kotłowski, Oluwasanmi Koyejo, and Nagarajan Natarajan. Consistency analysis for binary classification revisited. In *International Conference on Machine Learning*, pp. 961–969, 2017.

Elad ET Eban, Mariano Schain, Alan Mackey, Ariel Gordon, Rif A Saurous, and Gal Elidan. Scalable learning of non-decomposable objectives. *arXiv preprint arXiv:1608.04802*, 2016.

Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification. In *Proceedings of the IEEE international conference on computer vision*, pp. 1026–1034, 2015.

Sergey Ioffe and Christian Szegedy. Batch normalization: Accelerating deep network training by reducing internal covariate shift. *arXiv preprint arXiv:1502.03167*, 2015.

Martin Jansche. Maximum expected f-measure training of logistic regression models. In *Proceedings of the conference on Human Language Technology and Empirical Methods in Natural Language Processing*, pp. 692–699. Association for Computational Linguistics, 2005.

Thorsten Joachims. A support vector method for multivariate performance measures. In *Proceedings of the 22nd international conference on Machine learning*, pp. 377–384. ACM, 2005.

Purushottam Kar, Harikrishna Narasimhan, and Prateek Jain. Online and stochastic gradient methods for non-decomposable loss functions. In *Advances in Neural Information Processing Systems*, pp. 694–702, 2014.

Oluwasanmi O Koyejo, Nagarajan Natarajan, Pradeep K Ravikumar, and Inderjit S Dhillon. Consistent binary classification with generalized performance metrics. In *Advances in Neural Information Processing Systems*, pp. 2744–2752, 2014.

Alex Krizhevsky. Learning multiple layers of features from tiny images. Technical report, Citeseer, 2009.

Ye Nan, Kian Ming Chai, Wee Sun Lee, and Hai Leong Chieu. Optimizing f-measure: A tale of two approaches. *arXiv preprint arXiv:1206.4625*, 2012.

Harikrishna Narasimhan, Purushottam Kar, and Prateek Jain. Optimizing non-decomposable performance measures: a tale of two classes. In *International Conference on Machine Learning*, pp. 199–208, 2015.

Shameem Puthiya Parambath, Nicolas Usunier, and Yves Grandvalet. Optimizing f-measures by cost-sensitive classification. In *Advances in Neural Information Processing Systems*, pp. 2123–2131, 2014.

Joan Pastor-Pellicer, Francisco Zamora-Martínez, Salvador España-Boquera, and María José Castro-Bleda. F-measure as the error function to train neural networks. In *International Work-Conference on Artificial Neural Networks*, pp. 376–384. Springer, 2013.

Amartya Sanyal, Pawan Kumar, Purushottam Kar, Sanjay Chawla, and Fabrizio Sebastiani. Optimizing non-decomposable measures with deep networks. *arXiv preprint arXiv:1802.00086*, 2018.

Cornelis Joost Van Rijsbergen. Foundation of evaluation. *Journal of Documentation*, 30(4):365–373, 1974.

Vladimir Vapnik. Principles of risk minimization for learning theory. In *Advances in neural information processing systems*, pp. 831–838, 1992.

PROOFS

Proof. Theorem 1

Let $K = |\mathcal{Y}| \times |\mathcal{Y}|$. We enumerate the entries of C_D and C as $\Phi_{i,D}$ and Φ_i $(i = 1, ..., K)$. Let E_i $(i = 1, ..., K)$ denote the event that

$$\lim_{|D|\to\infty} \Phi_{i,D} = \Phi_i$$

and E denote the event that

$$\lim_{|D|\to\infty} F\big(\Phi_D\big) = F\big(\Phi\big)$$

We have

$$\bigcap_{i=1}^{K} E_i \subset E$$

because by continuity of F,

$$\Big(\forall i = 1, ..., K, \lim_{|D|\to\infty} \Phi_{i,D} = \Phi_i\Big) \Rightarrow \Big(\lim_{|D|\to\infty} \Phi_D = \Phi\Big)$$
$$\Rightarrow \Big(\lim_{|D|\to\infty} F\big(\Phi_D\big) = F\big(\Phi\big)\Big)$$

Taking complement on both sides, we have

$$E^c \subset \left(\bigcap_{i=1}^{K} E_i\right)^c = \bigcup_{i=1}^{K} E_i^c$$

By monotonicity of probability measure and union bound, we have

$$\mathbb{P}(E^c) \leq \mathbb{P}\left(\left(\bigcap_{i=1}^{K} E_i\right)^c\right) = \mathbb{P}\left(\bigcup_{i=1}^{K} E_i^c\right) \leq \sum_{i=1}^{K} \mathbb{P}(E_i^c) = \sum_{i=1}^{K} 1 - \mathbb{P}(E_i) = \sum_{i=1}^{K} 1 - 1 = 0$$

Consequently, we have

$$\mathbb{P}(E) = 1 - \mathbb{P}(E^c) = 1 - 0 = 1$$

i.e.

$$\mathbb{P}\Big(\lim_{|D|\to\infty} F\big(\Phi_D\big) = F\big(\Phi\big)\Big) = 1$$

□

Proof. Theorem 2

Let $K = |\mathcal{Y}| \times |\mathcal{Y}|$. As in the previous proof, we enumerate the entries of $\hat{C}_B(h_\theta)$ and $\hat{C}_D(h_\theta)$ as $\Phi_{i,B}(h_\theta)$ and $\Phi_i(h_\theta)$ $(i = 1, ..., K)$. Let E_i $(i = 1, ..., K)$ denote the event that

$$\lim_{|B|\to\infty} \nabla_\theta F\big(\Phi_{i,B}(h_\theta)\big) = \nabla_\theta F\big(\Phi_i(h_\theta)\big)$$

and E denote the event that

$$\lim_{|B|\to\infty} \nabla_\theta F\big(\Phi_B(h_\theta)\big) = \nabla_\theta F\big(\Phi(h_\theta)\big)$$

If we have

$$\bigcap_{i=1}^{K} E_i \subset E$$

which is equivalent to

$$\Big(\forall i = 1, ..., K, \lim_{|D|\to\infty} \nabla_\theta \Phi_{i,D}(h_\theta) = \nabla_\theta \Phi_i(h_\theta)\Big) \tag{10}$$

$$\Rightarrow \Big(\lim_{|D|\to\infty} \nabla_\theta F\big(\Phi_D(h_\theta)\big) = \nabla_\theta F\big(\Phi_i(h_\theta)\big)\Big) \tag{11}$$

then the theorem will follow from an argument similar to that in the proof of theorem 3.

We now prove this condition. By chain rule,

$$\nabla_\theta F\big(\Phi_D(h_\theta)\big) = \nabla F\big(\Phi_D(h_\theta)\big) \cdot \boldsymbol{J}_D$$

where $\nabla F\big(\Phi_D(h_\theta)\big)$ denotes the gradient of F w.r.t. $\Phi_D(h_\theta)$ and $\boldsymbol{J}(\theta)$ denotes the Jacobian of $\Phi_D \circ h$ w.r.t. θ.

By continuity of ∇F,

$$\begin{aligned}\Big(\forall i = 1, ..., K, \lim_{|D|\to\infty} \Phi_{i,D}(h_\theta) = \Phi_i(h_\theta)\Big) &\Rightarrow \Big(\lim_{|D|\to\infty} \Phi_D(h_\theta) = \Phi(h_\theta)\Big) \\ &\Rightarrow \Big(\lim_{|D|\to\infty} \nabla F\big(\Phi_D(h_\theta)\big) = \nabla F\big(\Phi(h_\theta)\big)\Big)\end{aligned}$$

Thus

$$\forall \delta, \exists N_{\nabla F,\delta}, |D| > N_{\nabla \boldsymbol{J},\delta} \Rightarrow |\boldsymbol{\epsilon}| = \big|\nabla F\big(\Phi_D(h_\theta)\big) - \nabla F\big(\Phi(h_\theta)\big)\big| < \frac{\delta}{3}$$

By definition of Jacobian,

$$\boldsymbol{J}_D(\theta) = \begin{bmatrix} \nabla_\theta \Phi_{1,D}(h_\theta) \\ ... \\ \nabla_\theta \Phi_{K,D}(h_\theta) \end{bmatrix}$$

Thus

$$\Big(\forall i = 1, ..., K, \lim_{|D|\to\infty} \nabla_\theta \Phi_{i,D}(h_\theta) = \nabla_\theta \Phi_i(h_\theta)\Big) \Rightarrow \Big(\lim_{|D|\to\infty} \boldsymbol{J}_D(\theta) = \boldsymbol{J}(\theta)\Big)$$

where

$$\boldsymbol{J}_D(\theta) = \begin{bmatrix} \nabla_\theta \Phi_D(h_\theta) \\ ... \\ \nabla_\theta \Phi_D(h_\theta) \end{bmatrix}$$

and the convergence is w.r.t. Euclidean norm, defined as

Thus

$$\forall \delta, \exists N_{\nabla F,\delta}, |D| > N_{\nabla \boldsymbol{J},\delta} \Rightarrow |\boldsymbol{\epsilon}| = \big|\big|\boldsymbol{J}_D(\theta) - \boldsymbol{J}(\theta)\big|\big| < \frac{\delta}{3}$$

where $||\cdot||$ denotes the matrix norm defined as

$$||\boldsymbol{A}|| = \sup_{|\boldsymbol{x}|\leq 1} |\boldsymbol{A}\boldsymbol{x}|$$

Thus for any D such that $|D| > \max\big(N_{\nabla F,\delta}, \exists N_{\nabla \boldsymbol{J},\delta}\big)$, by definition of matrix norm we have

$$\begin{aligned}\Big|\nabla_\theta F\big(\Phi_D(h_\theta)\big) - \nabla_\theta F\big(\Phi(h_\theta)\big)\Big| &= \Big|\nabla F\big(\Phi_D(h_\theta)\big) \cdot \boldsymbol{J}_D(\theta) - \nabla_\theta F\big(\Phi(h_\theta)\big)\Big| \\ &= \Big|\Big(\nabla F\big(\Phi(h_\theta)\big) + \boldsymbol{\epsilon}\Big)\Big(\boldsymbol{J}(\theta) + \mathcal{E}\Big) - \nabla_\theta F\big(\Phi(h_\theta)\big)\Big| \\ &= \Big|\nabla F\big(\Phi(h_\theta)\big)\boldsymbol{J}(\theta) + \nabla F\big(\Phi(h_\theta)\big) \cdot \mathcal{E} + \boldsymbol{\epsilon}\boldsymbol{J}(\theta) + \boldsymbol{\epsilon}\mathcal{E} - \nabla_\theta F\big(\Phi(h_\theta)\big)\Big| \\ &= \Big|\nabla F\big(\Phi(h_\theta)\big) \cdot \mathcal{E} + \boldsymbol{\epsilon}\boldsymbol{J}(\theta) + \boldsymbol{\epsilon}\mathcal{E}\Big| \\ &\leq \big|\nabla F\big(\Phi(h_\theta)\big) \cdot \mathcal{E}\big| + \big|\boldsymbol{\epsilon}\boldsymbol{J}(\theta)\big| + \big|\boldsymbol{\epsilon}\mathcal{E}\big| \\ &\leq \big|\nabla F\big(\Phi(h_\theta)\big)\big|\big|\mathcal{E}\big| + |\boldsymbol{\epsilon}|\big|\boldsymbol{J}(\theta)\big| + |\boldsymbol{\epsilon}|\big|\mathcal{E}\big| \\ &\leq \frac{\delta}{3} + \frac{\delta}{3} + \frac{\delta}{3} \\ &= \delta\end{aligned}$$

which implies that

$$\lim_{|D|\to\infty} \nabla_\theta F\big(\Phi_D(h_\theta)\big) = \nabla_\theta F\big(\Phi(h_\theta)\big)$$

□

Proof. Theorem 3

This theorem follows from Theorem 1.

□

Proof. Theorem 4

To prove consistency, it suffices to prove that (Vapnik (1992)):

$$\lim_{|D|\to\infty} \mathbb{P}\big(\sup_{\theta} \big|F(\hat{C}_D(h_\theta)) - F(\bar{C}(h_\theta))\big| < \epsilon\big) = 1$$

Because F is well behaved, by the union bound argument in previous proofs, it suffices to show that

$$\lim_{|D|\to\infty} \mathbb{P}\big(\sup_{\theta} \big|\big(\hat{C}_D\big)_{ij}(h_\theta) - \bar{C}_{ij}(h_\theta)\big| < \epsilon\big) = 1 \qquad 1 \leq i, j \leq |\mathcal{Y}|$$

which is true if p_θ has finite VC-dimension (see, for example, Koyejo et al. (2014)). The VC-dimension of p_θ can be limited to finite by various means. One way is to limit the outputs of p_θ to discrete values, e.g. all float point values between 0 and 1 that can be represented in a computer.

□

WELL BEHAVED PERFORMANCE METRICS

- accuracy $= \text{tp} + \text{tn}$
- F-measure
- Q-Mean $= 1 - \sqrt{\frac{(1-\text{tp})^2+(1-\text{tn})^2}{2}}$
- G-Mean $= \sqrt{\text{tp} \cdot \text{tn}}$
- Jaccard $= \frac{\text{tp}}{\text{tp}+\text{fp}+\text{fn}}$
- AUC $= \frac{\text{fp} \cdot \text{fn}}{(\text{tp}+\text{fn})(\text{fp}+\text{tn})}$
- Min $= \min\{\text{tp}, \text{tn}\}$

These performance metrics can be extended to multi-class classification in the same way that F-measure is extended to multi-class classification.