1 Abstract

- We present a method for multi-label classification that is able to optimise approximate surrogates for a variety of performance measures, such as macro-F$_\beta$, micro-precision and macro-recall.
- This means that we can tailor the objective function being optimised to the performance measure on which we want to do well in our specific application.
- Our method is based on well-understood facts from the domain of structured output learning, which gives us theoretical guarantees regarding the accuracy of the results obtained.
- Source code is available.

2 The Problem

In the usual multi-label classification setting we are given an instance (e.g. an image) and we want to predict its corresponding set of labels (e.g. tags).

However, most applications of multi-label classification are in information retrieval, where we are usually interested in the reverse problem: given a label, we want to predict a set of instances. This is reflected in the fact that the most widely used performance measures for multi-label classification are averaged over labels, and not over instances.

We therefore formulate the prediction problem in a reverse manner, in the spirit of [5]. Our approach is similar to [5], but it is a specialisation for cases in which the loss decomposes over the labels.

3 The Model

- \( x \in X \) denotes a label (e.g., a tag of an image)
- \( y \in \mathcal{Y} \) denotes a set of instances, (e.g., a set of images)
- \( N = |\mathcal{Y}| \) is the number of labels

An input label \( x \) is encoded as \( x \in \{0,1\}^N \), s.t. \( \sum y_i = 1 \). For example if \( N = 5 \) the second label is denoted as \( x = [0 1 0 0 0] \).

An output instance \( y \) is encoded as \( y \in \{0,1 \}^{|\mathcal{Y}|} \), and \( |\mathcal{Y}| = 1 \) iff instance \( y \) was associated with label \( x \).

We assume a given training set \( \{(x^i,y^i)\}_{i=1}^N \). The task consists of estimating a map \( f : X \to \mathcal{Y} \) which reproduces well the outputs of the training set \( f(x^i) = y^i \) but also generalises well to new test instances.

4 Performance Measures

- Macro-precision: \( \frac{1}{N} \sum_{y \in \mathcal{Y}} \frac{\sum_y \delta(y \to x)}{|\{y \in \mathcal{Y} : \delta(x \to y) \neq 0\}|} \)
- Macro-recall: \( \frac{1}{N} \sum_{x \in X} \frac{\sum_y \delta(y \to x)}{|\{y \in \mathcal{Y} : \delta(x \to y) \neq 0\}|} \)
- Macro-F$_\beta$: \( \frac{1}{N} \sum_{x \in X} \frac{\beta \cdot \sum_{y \in \delta(x \to y) \neq 0} \delta(y \to x)}{(1 + \beta^2) \cdot \sum_{y \in \delta(x \to y) \neq 0} \delta(y \to x)} \)

5 Loss Functions

The loss function represents how much we want to penalise a prediction \( \delta(x \to y) \) when the correct prediction is \( \delta(y \to x) \), i.e., it has the opposite semantics of a performance measure.

We can deal with a variety of loss functions in this framework, but for concreteness of exposition we focus on a loss derived from the macro-F$_\beta$ score, whose particular case for \( \beta = 1 \) is arguably the most popular performance measure for multi-label classification:

\[
\Delta(x, y) = 1 - \left(1 + \beta^2\right) \frac{\sum_{y \in \mathcal{Y}} \delta(y \to x)}{\sum_{y \in \mathcal{Y}} \delta(x \to y)}
\]

6 Features and Parameterization

We assume that the prediction for a given input \( x \) returns the maximiser(s) of a linear score of the model parameter vector \( \theta \), i.e., a prediction is given by \( \hat{y} = \delta(x \to y) \) such that

\[
\hat{y} = \arg\max_{y \in \mathcal{Y}} \langle \phi(x,y), \theta \rangle
\]

where we assume that \( \phi(x,y) \) is linearly composed of features of the instances encoded in each \( y \), i.e., \( \delta(x \to y) = \sum_{i=1}^N \phi_i(x,y) \). The vector \( \phi_i \) is the feature representation for the instance \( i \).

The map \( \phi(x,y) \) will be the zero vector whenever \( y \not\in \mathcal{Y} \), i.e., when instance \( x \) does not have label \( y \). The feature map \( \phi(x,y) \) has a total of \( DN \) dimensions, where \( D \) is the dimensionality of our instance features \( \phi(x) \) and \( N \) is the number of labels. Therefore \( DN \) is the dimensionality of our parameter vector \( \theta \) to be learned.

7 Optimisation Problem

Our ideal estimator has the form:

\[
\theta^* = \arg\min_{\theta} \left\{ \frac{1}{N} \sum_{x \in X} \Delta(x, \delta(x \to y)) + \frac{1}{2} \lambda \| \theta \|^2 \right\}
\]

This optimisation problem however is non-convex. We therefore use the approach popularised in [8], which optimises a convex upper bound on this structured loss:

\[
\bar{\theta} = \arg\min_{\theta} \left\{ \sum_{x \in X} \frac{1}{2} \sum_{y \in \mathcal{Y}} \delta(y \to x) \left( \frac{\gamma(x,y) - \Delta(x, y)}{\gamma(x,y)} \right)^2 + \frac{1}{2} \lambda \| \theta \|^2 \right\}
\]

8 Constraint generation

Due to the exponential number of constraints we resort to a constraint generation strategy, which consists of starting with no constraints and iteratively adding the most violated constraint for the current solution of the optimisation problem. The key problem that needs to be solved at each iteration is to find the maximiser of the violation margin \( \Delta_\nu \),

\[
\Delta_\nu = \max_{y \in \mathcal{Y}} (\langle \phi(x,y), \theta \rangle - \Delta(x, y)) \geq \Delta(x, y) - \ell(x, y), \ell(x, y) \geq 0
\]

9 Prediction at Test Time

The problem to be solved at test time (eq. (2)) has the same form as the problem of constraint generation (eq. (6)), the only difference being that \( \Delta_\nu = \langle \phi(x,y), \theta \rangle \) is not present.

Since \( \Delta_\nu \) is a constant vector, the solution \( \bar{\theta} \) for eq. (6) is the vector that indicates the positive entries of \( \Delta_\nu \), which can be efficiently found in \( O(DN) \) time. Therefore inference at prediction time is very fast.

11 Bibliography