A One-page Summary on *"Online Semi-supervised Multi-label Classification with Label Compression and Local Smooth Regression"*

Matin Moezzi
matin.moezzi@gmail.com

I. Introduction

Classification is a subcategory of supervised learning which aims to predict the categorical class labels of new instances based on past observations. If target labels are not mutually exclusive, which means each input instance can belong to more than one label class, we call this multi-label classification.

In many real-world applications, labeling training data is a prohibitively expensive and challenging task, especially in multi-label classification. Semi-supervised learning is a paradigm of learning which uses unlabeled data along with a small proportion of labeled data as supervision information to make predictions on new examples. Semi-supervised classification eliminates the problem of the scarcity of labeled data.

This paper presents a novel algorithm for online semi-supervised (inductive) multi-label classification, enabling real-time multi-label prediction in a semi-supervised setting and is robust to evolving the label space.

Prior to this work, there have been many successful efforts for addressing multi-label classification with static data and transductive semi-supervised classification. Despite these successes, transductive semi-supervised classification cannot be used in an online environment because this setting just predicts unlabeled training data, which may not cover all label distribution and cannot classify new unseen labels.

II. Problem Definition

This paper addresses multi-label classification tasks with three key considerations which are common properties of real-world applications, (1) Labeled and unlabeled training data arrive randomly in data streams over time, (2) The label space extends when a new label arrives, and (3) Predictions must occur in real-time.

III. Proposed Approach

The authors propose a framework named OnSeML which consists of three steps, label compression, local smooth regression, and an adaptive update which I elaborate on these in the following paragraphs to tackle the aforementioned problems.

A. Label Compression

First, OnSeML compresses the label set into a low-dimensional space to capture the high-order label relationship and to obtain a compact label space for the next step, the regression model. The one-hot label vector of the i-th instance $y_i \in \mathbb{B}^l$ (which l is the number of label classes) is encoded via an orthogonal encoding matrix $\mathbf{P} \in \mathbb{R}^{k \times l}$, $k < l$ ($\mathbf{h}_i = \mathbf{P}\mathbf{y}_i$). In the next step, the regression model $f_i : \mathbf{x}_i \rightarrow \hat{\mathbf{h}}_i$ estimates the encoded label for every instance whether it is labeled or not. After estimation, $\hat{\mathbf{h}}_i$ is decoded to the predicted label via the decoding matrix at time t, $\mathbf{Q}_t \in \mathbb{R}^{l \times k}$ ($\hat{\mathbf{y}}_i = \mathbf{Q}_t\hat{\mathbf{h}}_i$). The decoding matrix $\mathbf{Q}_t$ is obtained by minimizing the least square errors, which can be written as the following closed-form solution. $\mathbf{Q}_t = \mathbf{Y}_t\mathbf{H}_l^T(\mathbf{H}_t\mathbf{H}_l^T)^{-1}$ where $[\mathbf{X}_t, \mathbf{Y}_t]$ is a set of labeled instances and $\mathbf{H}_t = \mathbf{P}\mathbf{Y}_t$

B. Local Smooth Regression

To estimate the encoded label for each instance, a linear regression model is used as follows: $f_i(\mathbf{x}_i) = \mathbf{W}_i\mathbf{x}_i + \mathbf{b}_i$ where $\mathbf{W}_i \in \mathbb{R}^{k \times d}$ (d is the feature dimenstion) and $\mathbf{b}_i \in \mathbb{R}^k$ are the two parameters to be learned for the i-th instance. The authors use the core idea of regularized moving least square (RMLS) by approximating and regularizing the regression function for each incoming instance with its neighbors in the already arrived data using the kNN algorithm.

In order to prevent a linear increase of the model size with the number of all arriving instances, the model size is bounded by two buffers, B_L and B_A, for labeled and arrived points, respectively. Once the number of instances in these two buffers surpasses its predefined size, the oldest ones will be removed.

Another problem is the error propagation with more instances arriving. If choosing $\mathbf{x}_j$ is restricted to the $\mathbf{x}_i$ neighbors in the second term of the loss function, more related instances are selected, and propagated errors are reduced.

C. Adaptive Update

According to III-A, updating the decoding matrix $\mathbf{Q}_t$, which contains an inverse operator, each time a labeled instance arrives causes high computational cost. Therefore, OnSeML updates $\mathbf{Q}_t$ when the size of newly arrived labeled data reaches a pre-defined number.

In addition to the adaptive update, OnSeML uses an error adjustment module that estimates a new label decoder in an offline manner using recently arrived labeled data if the average prediction error is more than a specific error.