OCR learned by Structured Output SVM

UCU Summer school

July 13, 2017

1 Introduction

We will consider a problem of recognizing a given name from handwritten characters. The inputs are a binary images containing a line of text. All characters have a fixed width so that we do not have to solve the segmentation problem. A sample of the input images is shown in Figure 1. In Section 2 we will describe two linear classifiers for sequence recognition each of them using a different model of the sequences. Your task will be to learn parameters of these classifiers by Bundle Risk Minimization method and to evaluate their performance using data described in Section [?].

2 Structured output classifiers

The input is a binary image $\mathcal{I} = \{\mathcal{I}_1, ..., \mathcal{I}_L\} \in \{0,1\}^{16\times 8\cdot L}$ displaying a sequence of L handwritten characters. Each sub-image $\mathcal{I}_i \in \{0,1\}^{16\times 8}$, $i \in \{1, ..., L\}$ is of fixed size hence the segmentation of the image L into characters is known. We represent the input image $\mathcal{I}$ by a matrix $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_L) \in \mathbb{R}^{d\times L}$, where d is the number of features. The i-th column $\boldsymbol{x}_i \in \mathbb{R}^d$ is a feature description of the i-th sub-image $\mathcal{I}_i$.

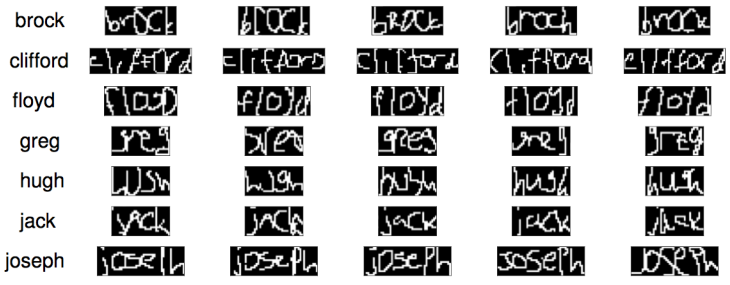

Figure 1: A sample of the input images for a selected sub-set of names

The task is to recognize a sequence of characters $y = (y_1, ..., y_L) \in \mathcal{Y}^L$, $\mathcal{Y} = \{a, b, ..., z\}$, depicted on the image $\mathcal{I}$ using the features $\boldsymbol{X}$. Below we describe different classification strategies and a way how to evaluate their performance.

2.1 Independent linear multi-class classifier

Given a feature representation $\boldsymbol{X} = (\boldsymbol{x}_1, ..., \boldsymbol{x}_L)$ of an input image $\mathcal{I}$, the predicted for each sub-image independently by a multi-class linear classifier $h\colon \mathcal{X} \to \mathcal{Y}$:

$$y^i = \operatorname*{argmax}_{y \in \mathcal{Y}} \langle \boldsymbol{w}_y, x^i \rangle \,, \forall i \in \{1, ..., L\} \,,$$

where the parameters are the character templates $\boldsymbol{w} = (\boldsymbol{w}_1, ..., \boldsymbol{w}_{|\mathcal{Y}|}) \in \mathbb{R}^{d \times |\mathcal{Y}|}$. You may consider adding biasses into $\boldsymbol{w}$. Note, the length L of the unknown sequence can be deduced from the width of the input image $\mathcal{I}$ or the feature matrix $\boldsymbol{X}$, respectively.

2.2 Linear structured output classifier

A linear classifier over sequences $h\colon \mathcal{I}^L \to \mathcal{Y}^L$

$$(\hat{y}_1, ..., \hat{y}_L) = \operatorname*{argmax}_{(y_1, ..., y_L) \in \mathcal{Y}^{\mathcal{L}}} \sum_{i=1}^{L} q(x_i, y_i) + \sum_{i=1}^{L-1} g(y_i, y_{i+1}) \tag{1}$$

where $q(x^i, y^i) = \langle \boldsymbol{w}^q_{y^i}, x^i \rangle$ and $g(y^i, y^{i+1}) = \boldsymbol{w}^g_{y^i, y^{i+1}}$, where $\boldsymbol{w}^q = (\boldsymbol{w}_1, ..., \boldsymbol{w}_{|\mathcal{Y}|}) \in \mathbb{R}^{d \times |\mathcal{Y}|}$ and $\boldsymbol{w}^g \in \mathbb{R}^{|\mathcal{Y}| \times |\mathcal{Y}|}$. In this class the pairwise function for classifier can be written in the matrix form $q(\boldsymbol{w}, x) := \boldsymbol{W}^q x$, where $\boldsymbol{W}^q$ is vector $(\boldsymbol{w}_1, ..., \boldsymbol{w}_{|Y|})$ stacked row-wise. Pairwise function is simply $g := \boldsymbol{W}^g$,

For begin short we denoted (1) as $\langle \boldsymbol{w}, \phi(x^i, y^i) \rangle$ in the lectures.

3 The task

In this task you have to implement surrogate losses

- for multi-class linear classifier:

$$\psi(x^i, y^i, \boldsymbol{w}) = \max_{y \in \mathcal{Y}} \{ [\![y^i \neq y]\!] + \langle \boldsymbol{w}_y, x^i \rangle - \langle \boldsymbol{w}_{y^i}, x^i \rangle \}$$

- for structured-output classifier on the sequences:

$$\psi(x^i, y^i, \boldsymbol{w}) = \max_{y \in \mathcal{Y}} \sum_{k=1}^{L} [\![y^i_k \neq y_k]\!] + \langle \boldsymbol{w}, \phi(x^i, y) \rangle - \langle \boldsymbol{w}, \phi(x^i, y^i) \rangle \,,$$

for learning linear multi-class and structured output classifiers respectively using the Bundle Method for Risk Minimization (BMRM) solver. It accesses

manually specified surrogate loss as first order oracle. That means, to use BMRM you have to provide a python class with overloaded method $\_\_call\_\_$ that takes numpy vector $\boldsymbol{w} \in \mathbb{R}^{n \times 1}$ and returns pair: (surrogate risk, subgradient) for given input $\boldsymbol{w}$. We provide you a python implementation of BMRM.

Useful hint for both tasks.

The function $F(\boldsymbol{w}) = \max\limits_{i=1,\ldots,n} (A\boldsymbol{w})$, where $A \in \mathbb{R}^{n \times n}$, has subgradient a_i^T, where a_i^T is the i-th row of the matrix on which maximum is achieved.