STATISTICAL MACHINE LEARNING (WS2022)
COMPUTER LAB
OCR learned by Structured Output Perceptron

VF

1 Introduction

We will consider a problem of recognizing a given name from images of 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. A relative frequency of the names is in Figure 2.

In Section 2 we will describe three linear classifiers for sequence recognition, each using a different model of the sequences. Your task will be to learn parameters of the three classifiers by the Perceptron algorithm and to evaluate their performance. Links to the data are provided in Section 3. The assignments to solve are described in Section 4.

2 Structured output classifiers

The input is a binary image $\mathcal{I} = (\mathcal{I}_1, \ldots, \mathcal{I}_L) \in \{0,1\}^{16\times 8\cdot L}$ displaying a sequence of L handwritten characters. Each sub-image $\mathcal{I}_i \in \{0,1\}^{16\times 8}$, $i \in \{1,\ldots,L\}$, is of fixed size hence the segmentation of the image $\mathcal{I}$ into characters is known. We represent the input image $\mathcal{I}$ by a matrix $\boldsymbol{X} = (\boldsymbol{x}_1,\ldots,\boldsymbol{x}_L) \in \mathbb{R}^{d\times L}$ 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 $\mathcal{I}_i$. The feature vector $\boldsymbol{x}_i$ contains the intensity values of $\mathcal{I}_i$ and un-ordered products of the intensity values computed from all different pixel pairs of $\mathcal{I}_i$. Therefore the number of features is $d = 16\cdot 8 + (16\cdot 8 - 1)(16\cdot 8)/2 = 8256$. In addition, each feature vector is normalized to have a unit L_2-norm.

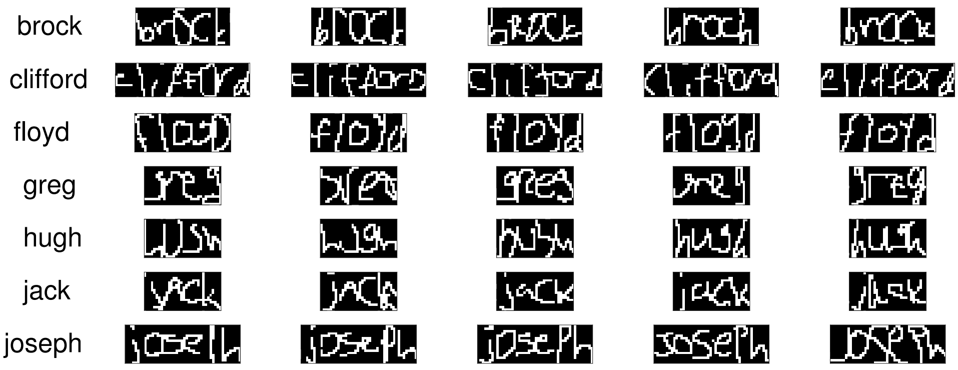

Figure 1: A sample of the input images for a selected sub-set of names.

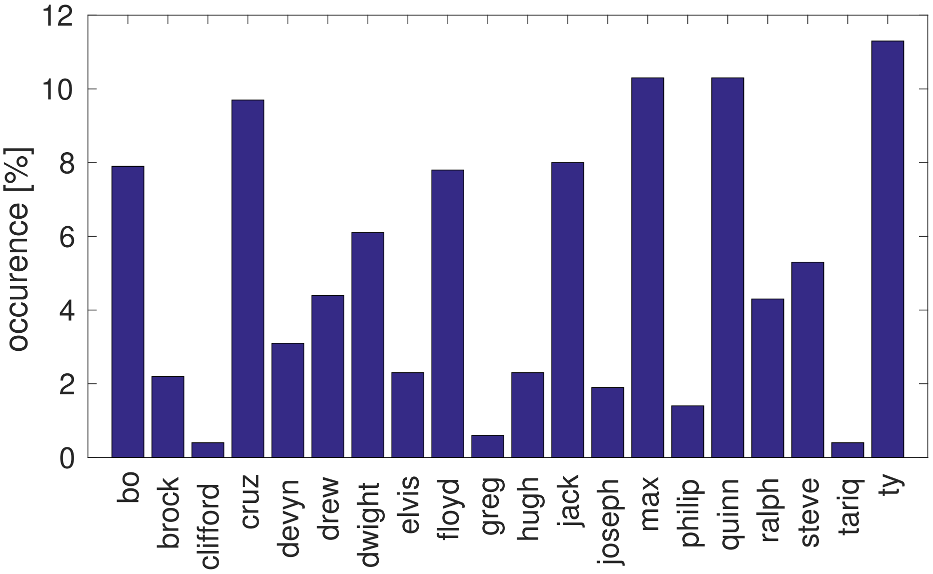

Figure 2: The relative frequency of the names in the training and testing dataset. The names selected to the training/testing dataset form a subset of most popular given names in USA used in 1990. The names and the frequencies were selected so that the occurrence of individual characters in the training/testing dataset is close to the uniform distribution.

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

2.1 Independent linear multi-class classifier

Given a feature representation $\boldsymbol{X} = (\boldsymbol{x}_1, \ldots, \boldsymbol{x}_L)$ of an input image $\mathcal{I}$, the characters are predicted for each sub-image independently by a multi-class linear classifier

$$\hat{y}_i \in \operatorname*{Argmax}_{y \in \mathcal{A}} (\langle \boldsymbol{w}_y, \boldsymbol{x}_i \rangle + b_y) \,, \qquad i \in \{1, \ldots, L\} \,, \tag{1}$$

where the parameters are the character templates $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{|\mathcal{A}|}) \in \mathbb{R}^{d \cdot |\mathcal{A}|}$ and biases $\boldsymbol{b} = (b_1, \ldots, b_{|\mathcal{A}|}) \in \mathbb{R}^{|\mathcal{A}|}$. Note that 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 classifier modeling pair-wise dependency

The characters are predicted by a linear classifier

$$(\hat{y}_1, \ldots, \hat{y}_L) \in \operatorname*{Argmax}_{(y_1, \ldots, y_L) \in \mathcal{A}^L} \left(\sum_{i=1}^{L} (\langle \boldsymbol{w}_{y_i}, \boldsymbol{x}_i \rangle + b_{y_i}) + \sum_{i=1}^{L-1} g(y_i, y_{i+1}) \right) \tag{2}$$

where the parameters are $\boldsymbol{w} = (\boldsymbol{w}_1, \ldots, \boldsymbol{w}_{|\mathcal{A}|}) \in \mathbb{R}^{d \cdot |\mathcal{A}|}$, $\boldsymbol{b} = (b_1, \ldots, b_{|\mathcal{A}|}) \in \mathbb{R}^{|\mathcal{A}|}$ and the pair-wise dependency function $g \colon \mathcal{A} \times \mathcal{A} \to \mathbb{R}$.

Evaluating the predictor (2) leads to a discrete optimization problem which can be solved in time $\mathcal{O}(|\mathcal{A}|^2 \cdot L)$ by dynamic programming. By introducing a shortcut $q_i(y_i) = \langle \boldsymbol{w}_{y_i}, \boldsymbol{x}_i \rangle + b_{y_i}$, we can rewrite (2) as

$$(\hat{y}_1,\ldots,\hat{y}_L) \in \operatorname*{Argmax}_{(y_1,\ldots,y_L)\in\mathcal{Y}^L} \left(\sum_{i=1}^{L} q_i(y_i) + \sum_{i=1}^{L-1} g(y_i,y_{i+1}) \right) \tag{3}$$

Let $F_i\colon \mathcal{A} \to \mathbb{R}$, $i \in \{1,\ldots,L\}$, and $Y_i\colon \mathcal{A} \to \mathcal{A}$, $i \in \{2,\ldots,L\}$, be functions defined recursively as follows:

$$\begin{array}{rcll} F_1(y) & = & q_1(y)\,, & \forall y \in \mathcal{A}\,, \\ F_i(y) & = & q_i(y) + \max\limits_{y'\in\mathcal{A}} \Big(F_{i-1}(y') + g(y',y) \Big)\,, & i = 2,\ldots,L \\ Y_i(y) & \in & \operatorname*{Argmax}\limits_{y'\in\mathcal{A}} \Big(F_{i-1}(y') + g(y',y) \Big)\,, & i = 2,\ldots,L \end{array}$$

Then, the optimal value of the problem (3) equals to

$$F^* = \max_{y\in\mathcal{A}} F_L(y)\,,$$

and the last label in the optimal sequence is

$$\hat{y}_L \in \operatorname*{Argmax}_{y\in\mathcal{A}} F_L(y)\,.$$

The other labels in the optimal sequence are found recursively as follows

$$\hat{y}_{i-1} = Y_i(\hat{y}_i)\,, \qquad i = L-1,\ldots,2\,.$$

2.3 Linear structured classifier for fixed number of sequences

Let us assume that the set of hidden sequences $\mathcal{Y} \subset \mathcal{A}^*$ contains a small number of elements. For example, in our application $\mathcal{Y}$ contains just 20 names (see Figure 2). In this case we can predict the sequences by a linear classifier

$$(\hat{y}_1,\ldots,\hat{y}_L) \in \operatorname*{Argmax}_{(y_1,\ldots,y_L)\in\mathcal{Y}_L} \left(\sum_{i=1}^{L} (\langle \boldsymbol{w}_{y_i}, \boldsymbol{x}_i \rangle + b_{y_i}) + v(y_1,\ldots,y_L) \right) \tag{4}$$

where $\mathcal{Y}_L \subset \mathcal{Y}$ contains all sequences of the length L. The classifier is parametrized by $\boldsymbol{w} = (\boldsymbol{w}_1,\ldots,\boldsymbol{w}_{|\mathcal{A}|}) \in \mathbb{R}^{d\cdot|\mathcal{A}|}$, $\boldsymbol{b} = (b_1,\ldots,b_{|\mathcal{A}|}) \in \mathbb{R}^{|\mathcal{A}|}$ and a function $v\colon \mathcal{Y} \to \mathbb{R}$.

2.4 Error measures

Let $\{(\boldsymbol{X}^1,\boldsymbol{y}^1),\ldots,(\boldsymbol{X}^m,\boldsymbol{y}^m)\}$ be a set of examples and let $\{\hat{\boldsymbol{y}}^1,\ldots,\hat{\boldsymbol{y}}^m\}$ be the predictions produced by a classifier applied on the inputs $\{\boldsymbol{X}^1,\ldots,\boldsymbol{X}^m\}$. To evaluate a performance of a classifier we are going to use two error measures. First, the sequence prediction error defined as

$$R^{\text{seq}} = \frac{1}{m} \sum_{j=1}^{m} [\![\boldsymbol{y}^j \neq \hat{\boldsymbol{y}}^j]\!]$$

which is an estimate of the probability that the predicted sequence is not entirely correct. Second, the character prediction error is defined as

$$R^{\text{char}} = \frac{1}{M} \sum_{j=1}^{m} \sum_{i=1}^{L_j} [\![y_i^j \neq \hat{y}_i^j]\!]$$

where $M = \sum_{j=1}^{m} L_j$ is the total number of characters in all sequences and L_j is the length of the j-th sequence. The value of R^{char} is an estimate of the probability that a single character in the sequence is incorrectly classified.

3 Data

- The training and testing images `http://cmp.felk.cvut.cz/cmp/courses/SSU/sosvm_ocr/ocr_names_images.zip`
- Jupyter notebook which shows how to load images and compute the features `http://cmp.felk.cvut.cz/cmp/courses/SSU/sosvm_ocr/load_data_example.ipynb`
- The Matlab MAT file containing the data `http://cmp.felk.cvut.cz/cmp/courses/SSU/sosvm_ocr/ocr_names.mat`

The data were generated from the OCR benchmark described in [1].

4 Assignments

Assignment 1 (3 points) *Implement the Perceptron algorithm for learning parameters* ($\boldsymbol{w} \in \mathbb{R}^{d \cdot |\mathcal{A}|}, \boldsymbol{b} \in \mathbb{R}^{|\mathcal{A}|}$) *of the linear multi-class classifier (1). Use the provided training examples* $\mathcal{T}^m$ *to learn parameters of the classifier. Report the sequence prediction error* R^{seq} *and the character prediction error* R^{char} *computed on the provided testing examples* $\mathcal{S}^l$. *The output should be a single script (Jupyter notebook or Matlab) which learns the classifier and prints the computed testing errors.*

Assignment 2 (3 points) *Implement the Perceptron algorithm for learning parameters* ($\boldsymbol{w} \in \mathbb{R}^{d \cdot |\mathcal{A}|}, \boldsymbol{b} \in \mathbb{R}^{|\mathcal{A}|}, g \in \mathbb{R}^{|\mathcal{A}|^2}$) *of the linear structured output classifier (2). Evaluate the algorithm as specified in Assignment 1.*

Assignment 3 (3 points) *Implement the Perceptron algorithm for learning parameters* ($\boldsymbol{w} \in \mathbb{R}^{d \cdot |\mathcal{A}|}, \boldsymbol{b} \in \mathbb{R}^{|\mathcal{A}|}, v \in \mathbb{R}^{|\mathcal{Y}|}$) *of the linear structured output classifier (4). Evaluate the algorithm as specified in Assignment 1.*

Remark: The training examples are linearly separable with respect to the used features and all three linear classifiers (1), (2) and (4). This guarantees that the Perceptron algorithm will converge in a finite number of iterations. Use the zero training error as the stopping condition of the Perceptron algorithm. After convergence evaluate the training error in order to have a sanity check that your code is working properly.

Assignment 4 (1 point) *Summarize the testing errors of the three learned classifiers in a single table. For example, the summary table can have the following format:*

	Testing errors in %	
	R^{seq}	R^{char}
independent multi-class classifier	*TBA*	*TBA*
structured, pair-wise dependency	*TBA*	*TBA*
structured, fixed number of sequences	*TBA*	*TBA*

Explain differences in the performance of the three classifiers. Point out the main advantages and disadvantages of each classification model.

Assignment 5 (5 bonus points) *Propose and implement a classifier which has a better performance than the classifiers from Assignment 1,2 and 3. Describe your classifier in the report and include its performance on the testing examples. For instance, you can invent better features to describe the images, or you can invent a better structured classifier, or you can use the Structured Output SVM for learning or use any other idea.*

References

[1] B. Taskar, C. Guestrin, and D. Koller. Max-margin markov networks. In *NIPS*, volume 16. MIT Press, 2004.