EE660 Final Project: Optical Recognition of Handwritten Digits

Haiqiang Wang

December 8, 2015

Abstract

Optical Recognition of Handwritten Digits has been widely used in many pratical applications, it aims at converting images containing handwritten digits or printed digits into computer readable text. The difficulity lies in finding robust and effective digits preprocessing and segmentation algorithms. Digits in images are hard to retrieve due to the variability of digit size, handwritten styles, noisy background. Given a well-preprocessed dataset, where the binary image of a digit block is well represented, this report studys the performance of different machine learning approaches with the help of publicly available machine tool, LIBSVM. The classification accuracy achieves 98.27 % with RBF kernels and parameter optimization.

1 Project Homepage

`https://github.com/haiqianw/EE660_project`

2 Problem Statement and Preprocessing

The Optical Recognition of Handwritten Digits Data Set [1] was used in this report. Normalized bitmaps of handwritten digits from a preprinted form was available. These data are from 43 people, 30 of them contributed to the training set and another 13 to the test set.

Fig 1 gives two examples of handwritten digits, the objective is to convert handwritten digits to pritened digits, i.e. recognizing the first row as $7, 2, 1, 0, 4, 1, 4, 9, 5, 9$ in Fig 1a. An even challenging example was given in Fig 1b, digits are in regular shape due to the written style of a specific user and the limitation of image resolution.

Considerable efforts [2] are need in order to finish such a task, normally it involves lots of sophisticated preprocessing methodes organized in a hierarchical structure, which helps to divide a complicated task into several independent small tasks.

1. The bottom module is to adjust the contrast of the given images, usually this module is implentated as adaptive thresholding or *histogram eaqualization*.

2. The following module is about segmentation, humans have no difficulty in deciding which image block corresponds to a digit, but this process is not trival for computer. *Connected Components Analysis* (CCA) is a suitable technique to find the connected area of digits and then segment input image into multiple blocks, where each block contains a single digit.

3. The binarization process is to convert pixels in gray level i.e. $[0, 255]$ to binary representation $[0, 1]$, where 0 corresponds an empty pixel and 1 indicates the pixel is used to represent a digit.

(a)

(b)

Figure 1: Illustration of the handwritten digits recognition problem.

An example of Bitmap representation was given in fig 2 after the aforementioned precessing steps. In order to reduce the dimensionality of feature space, $32 * 32$ bitmaps are divided into nonoverlapping blocks of $4 * 4$. The number of pixels is counted in each block,

this leads to a feature in dimension of 64, and each contribute is an integer in the range of $0, \ldots, 16$.

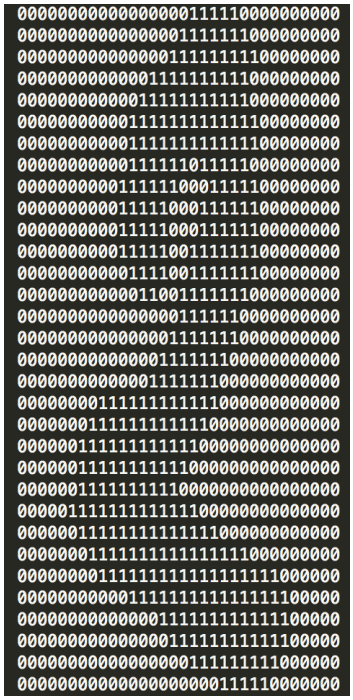

Figure 2: Illustration of handwritten digits recogintion

3 Prior and Related Work

Prior work invovles the preprocessing procedure, histogram equalization, connected components analysis, normalization and feature extraction. In Fall2013, I have finished a homework about Optical Character Recognition (OCR). In that project, I approached everything from the beginning, however, hand-crafter decision tree was used based on the properties of each character, i.e. Eular Number, Symmetry, Aspect Ratio, etc. It was a heuristic approach which had many hard thresholding. This project means to test the performance of standard machine learning approaches. The emphasis lies on the understanding and implentation of these approaches.

4 Project Formulation and Setup

In this project, handwritten digits recognition was approached as a multiclass classification problem. Each instance has a feature vector with dimension 64, and each element in the feature vector is an integer in the range $[0, 16]$, i.e. the number of pixels used to represent

digits. The class number is 10, ranging from $0, \cdots, 9$ as it is a digits recognition problem. The class lables are the true digit value of a specific handwritten digit. In the training process, both the instance and corresponding label are available. However, the testing set only has the instance available and the true labels are used to compute the performance of classification algorithms.

Support Vector Machines (SVMs) are used to achieve the classification task. Given a set of labeled instance $\boldsymbol{x}_i, y_i, i = 1, \ldots, M$ where $\boldsymbol{x}_i \in R^{64}$ and $y \in \{0, \ldots, 9\}^M$, the SVM solves the following optimization problem:

$$\begin{aligned} \min_{\boldsymbol{w},\boldsymbol{b},\boldsymbol{\xi}} \quad & \frac{1}{2}\boldsymbol{w}^T\boldsymbol{w} + C\sum_{i=1}^{M}\xi_i \\ s.t. \quad & y_i(\boldsymbol{w}^T\phi(\boldsymbol{x}_i) + b) \leq 1 - \xi_i, \\ and \quad & \xi_i \leq 0 \end{aligned}$$

In order to achieve a better performance, the training vectors $\boldsymbol{x}_i$ are mapped into a higher dimensional space through the *kernel trick* ϕ. SVM optimizes to separate the maximal margin by a hyperplane in the mapped high dimensional space. Parameter $C > 0$ is the penalty of the error term. According to our lecture and homework, we know that $K(\boldsymbol{x}_i, \boldsymbol{x}_j) \equiv \phi(\boldsymbol{x}_i)^T\phi(\boldsymbol{x}_j)$. In this project the following three kernels was used to compare their performance:

- linear: $K(\boldsymbol{x}_i, \boldsymbol{x}_j = \boldsymbol{x}_i^T\boldsymbol{x}_j)$
- polynomial: $K(\boldsymbol{x}_i, \boldsymbol{x}_j = (\gamma\boldsymbol{x}_i^T\boldsymbol{x}_j + r)^d, \gamma > 0$
- RBF: $K(\boldsymbol{x}_i, \boldsymbol{x}_j = exp(-\gamma||\boldsymbol{x}_i - \boldsymbol{x}_j||^2)$

Here, γ, r and d are kernel parameters to tune in order to assess their influence on performance.

5 Methodology

The data was randomly separated as training and testing set. The training and testing set has 3823 and 1797 samples, respectively. The hptothesis set is $H = \{0, \ldots, 9\}$, it is an one-vs-all classification problem. Before the training and testing procedure, feature vector was normalized to $[-1, 1]$ to avoid attirbutes in greater numeric ranges dominating those in smaller numeric ranges. In this project, each attribute was scaled to the range $[-1, +1]$.

For RBF kernels, 5-fold corss-validation and grid search was used to avoid overfitting or underfitting issues, it is essential to identify good (C, γ) in RBF kernel so that classifier can accurately predict unknown data. The training set was divided into 5 subsets of eauql size. Sequentially one subset was used as testing set using the classifier trained on the remaining 4 subsets. Hence, each instance in the training set was used only once as testing, the average accuracy of each testing was used to show the performance of parameters setting.

A straightforward grid search procedure was used in cross-validation. The parameter set which gives the best performance was selected as the final configuration. Some advanced optiminzation algorithms are available toward this end, however grid search was adopted since it is suitable for parallelization and the computation load is almost as much as those advanced algorithms. A coarse-finer grid search was used to alleviate the computation issue, a coarse grid was used first at the beginning, then a finer grid around the best selected coarse grid follows, this process terminates once some stop criteria was satisfied.

5.1 Implementation

My implementation was mainly based on LIBSVM [3], where many useful machine learning tools were intergrated in it. It has been widely used by researchers in the community of machine learning because of their excellent generalization for different tasks.

5.2 Feature Space

The dataset used in this project is *Optical Recognition of Handwritten Digits Data Set* [2]. The feature is with dimension of 64, each element is in the range of $[0, \ldots, 16]$, and the class label is between $[0, \ldots, 9]$. Their meanings are explained as follow, for a $32 * 32$ normalized bitmap, it was divided into unoverlapping $4 * 4$ blocks, the count of $1's$ in each block corresponds to an element of the feature.

5.3 Training, Validation and Model Selection

This project means to test the performance of different kernels in a multiclass classification problem. First, the whole database was randomly divided as training and testing set. In order to optimize parameters configuration on RBF kernels, a 5-fold cross-validation process and grid search were used. This also helps to avoid overfitting. Their performance was illustrated in Table 1.

Table 1: Classification performance comparison, where *Linear, Poly, RBF* stands for Linear kernel, Polynomial Kernel, and RBF Kernel. For Polynimial kernels, $degree = \{3, 4\}$ were tested.

	Linear	Poly (3)	Poly (4)	RBF	RBF+CV
PRE	95.99%	97.22%	96.93%	96.94%	**98.27**%

The best performance achieved is **98.27** as shown above.

6 Summary and Conclusions

This project studies the performance of different kernels on Optical Handwritten Recognition. Compared with other Computer Vision problem, the dataset is clean and well-organized, so the final performance achieves as high as 98.27%, the misclassified cases are

also difficult to recognize for human since they are written in unusual style. Generally, handwritten digits/texts recognition approaches failed to give a competitive performance, the reason is that the preprocessing approaches couldn't give a bitmap as clean as the provided one, the real battle ground is about accurately locating and segment each character automatically.

However, it is still an ideal case to study the performance of different kernels in machine learning without the influence of other preprocessing-related procedure. RBF kernel shows its nonlinear mapping ability over linear and polynomial kernels, which are special cases of RBF. Nonlinearly mapping the feature into a higher dimensional space helps to reveal the underlying nonlinearlity of feature.

References

[1] E. Alpaydin and C. Kaynak, "Cascading classifiers," *Kybernetika* **34**(4), pp. 369–374, 1998.

[2] M. D. Garris, J. L. Blue, G. T. Candela, *et al.*, "Nist form-based handprint recognition system (release 2.0)," 1997.

[3] C.-C. Chang and C.-J. Lin, "Libsvm: A library for support vector machines," *ACM Transactions on Intelligent Systems and Technology (TIST)* **2**(3), p. 27, 2011.