Midway Report: Semi-Supervised Learning of Pen-Based OCR.

Joshua Brakensiek, Jacob Imola, Sidhanth Mohanty

1 Background

Handwriting recognition is a classic machine learning problem. Generally, an image of a digit is inputted to an algorithm and classified. Instead of taking this approach, we used two data sets, "Pen-Based Recognition of Handwritten Digits Data Set" [AA98] and "UJI Pen Characters (Version 2) Data Set" [LPM$^+$08] from the UCI Machine Learning Repository [Lic13], which consist of pixel coordinates that the writers' pens took at certain time intervals. Each data set has over 10,000 data points. The first data set consists entirely of drawings of the digits $0, \dots, 9$, while the second data set uses a much richer variety of characters. We seek to explore if adding the extra information of time while sacrificing some detail from the shape of the digits can produce a viable semi-supervised model. We implement our learning algorithms in Python 2.x, using on the NumPy, SciPy, and Scikit-learn libraries (and possibly other libraries as we see fit). The models we try are Transductive Support Vector Machines and graph-based kernel methods and regularization. Throughout this paper, let $X = \{x_1, x_2, \dots, x_k\}$ be the labeled training data, $Y = \{y_1, y_2, \dots, y_k\}$ be the corresponding labels, $X^* = \{x_1^*, x_2^*, \dots, x_n^*$ be the unlabeled training data, and $Y^* = \{y_1^*, y_2^*, \dots, y_n^*$ be variables representing the unlabeled training data's labels. Now, we will give a description of the models:

Recall that an SVM reduces to the following primal problem:

$$\arg\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{k} \zeta_i \tag{1}$$

$$\begin{aligned} \text{subject to} \quad & \forall_{i=1}^{k} : y_i(wx_i + b) \geq 1 - \zeta_i \\ & \forall_{i=1}^{k} : \zeta_i > 0 \end{aligned}$$

A TSVM's primal problem allows the unlabeled data to take on any label, and hence reduces to a similar problem [Joa99]:

$$\arg\min_{w,b,\zeta,\zeta^*,Y^*} \frac{1}{2} w^T w + C \sum_{i=1}^{k} \zeta_i + C^* \sum_{i=1}^{n} \zeta_i^* \tag{2}$$

$$\begin{aligned} \text{subject to} \quad & \forall_{i=1}^{k} : y_i(wx_i + b) \geq 1 - \zeta_i \\ & \forall_{i=1}^{n} : y_i^*(wx_i^* + b) \geq 1 - \zeta_i^* \\ & \forall_{i=1}^{k} : \zeta_i > 0 \\ & \forall_{i=1}^{n} : \zeta_i^* > 0 \end{aligned}$$

Here, C and C^* are parameters that control the type of fit we get.

2 Related Work

(Probably could cite some of our Bibliography here)

3 Methods

One TSVM method we use is an SVM with local search which was first described in [Joa99]. We use a more generalized version of an SVM:

$$\arg\min_{w,b,\zeta,\zeta^*} \frac{1}{2}w^T w + C\sum_{i=1}^{k}\zeta_i + C^*_+\sum_{i=1}^{n}\zeta^*_i[y^*_i == 1] + C^*_-\sum_{i=1}^{n}\zeta^*_i[y^*_i == -1] \tag{3}$$

$$\begin{aligned} \text{subject to} \quad & \forall_{i=1}^{k} : y_i(wx_i + b) \geq 1 - \zeta_i \\ & \forall_{i=1}^{n} : y^*_i(wx^*_i + b) \geq 1 - \zeta^*_i \\ & \forall_{i=1}^{k} : \zeta_i > 0 \\ & \forall_{i=1}^{n} : \zeta^*_i > 0 \end{aligned}$$

Let's denote an SVM with such an objective function as $SVM(C, C^*_+, C^*_-, T^*)$. The user inputs C and C^* defined in (2) and two additional paremeters: num_+, which is the number of Y^* that will be equal to 1, and ϵ which is a weight to be used later. First, a regular SVM is trained with X, Y, and C, using the objective function of (1), and we find the margin distances of X^*. We take the num_+ most positive margin distances and classify them as 1 in Y^*. Then, we initialize two weights, $C^*_- = \epsilon$ and C^*_+ such that $\frac{C^*_+}{C^*_-} = \frac{num_+}{n - num_+}$. Then, we call $SVM(C, C^*_+, C^*_-, Y^*)$, and we greedily find two indices i and j such that $y^*_j = -y^*_i$, $\zeta^*_i > 0$ and $\zeta^*_j > 0$ and $\zeta^*_i + \zeta^*_j > 2$, then we know that flipping y^*_i and y^*_j will reduce (3) and preserve the number of positive num_+ examples. We do this as much as possible, and then set $C^*_+ = \max\{2C^*_+, C^*_+\}$ and $C^*_- = \max\{2C^*_-, C^*_-\}$. This increases the importance of the unlabeled data, because its label accuracy should be increasing. Finally, we call $SVM(C, C^*_+, C^*_-, Y^*)$ again, with the updated values of C^*_+, C^*_-, and Y^*, and repeat until $C^*_- = C^*_+ = C^*$.

4 Experiments

We tested the searching algorithm used in [Joa99] on the dataset found in [AA98]. We trained 10 different TSVMs, one to distinguish one digit from the rest of the data. To classify a given test point, we found the signed distance from the point to the decision boundary of each TSVM and found the maximum. We tried using three different kernels for our TSVM: linear, Gaussian, and sigmoid. The Gaussian and sigmoid kernels caused our TSVMs to predict all test points as -1, achieving 90% accuracy. We believe this is because the two kernels can express functions that are too complicated for the noise and high-dimension of this data set. We had more luck with the simple linear kernel, and some results are plotted in the tables below:

[Insert table with a few of the classification accuracies of the TSVMs]

Finally, we produced a noticeable relationship between percent of labeled data and classification accuracy

[Insert graph of classification accuracy vs percent of labeled data]

[Insert comments about digits that the TSVM often mistook for another]

5 Future Plans

Do feature selection on the data. Try different kernels for the TSVM, something more advanced than a linear kernel and not as advanced as a Gaussian kernel. Doing feature selection could help us find a good kernel. Change the algorithm of [Joa99] by not enforcing exactly num_+ examples to be classified as +1. Try different values for the parameters in the algorithm. Try more methods.

References

[AA98] E Alpaydin and Fevzi Alimoglu. Uci pen-based recognition of handwritten digits data set. 1998.

[Joa99] T Joachims. Transductive inference for text classifcation using support vector machines. 1999.

[Lic13] M. Lichman. UCI machine learning repository, 2013.

[LPM$^+$08] David Llorens, Federico Prat, Andrés Marzal, Juan Miguel Vilar, María José Castro, Juan-Carlos Amengual, Sergio Barrachina, Antonio Castellanos, Salvador España Boquera, JA Gómez, et al. The ujipenchars database: a pen-based database of isolated handwritten characters. In *LREC*, 2008.