Spring 2015 M 362K Self-Guided Project Proposal

Chen, Xiaohui
EID: xc2388

Meger, Sarah
EID: scm2595

Monsen, Brian
EID: brm756

March 3, 2015

In this project we will extend a coding assignment motivated by Dan Klein and Jogn DeNero given in UC Berkeley's CS188 course.

The coding assignment is to implement and design three classifiers: a naive Bayes classifier, a perceptron classifier and a large-margin (MIRA) classifier. We have around a thousand images of hand-written digits (0 to 9). The three classifiers we implemented will first analyze the images and create models correspondingly. After the three models are created, another set of images (around 100 images) will be the input and each model will figure out what each of the hand-written images is. After that we will compare the output with the correct values and figure out the rate of correctness of each models. In this case, we can figure out which classifier is the most optimal one.

The naive Bayes classifier uses Bayes' Theorem we learned in class. Let Y be the label and $F_1, F_2, \ldots, F_n$ be a set of observed random variables called features. It follows that $Pr(F_1, F_2, \ldots, F_n, Y) = Pr(Y)\prod\limits_i P(F_i|Y)$. Then $Pr(y|f_1, \ldots, f_m) = \frac{Pr(y)\prod_{i=1}^{m} Pr(f_i|y)}{Pr(f_1, \ldots, f_m)}$. In order to determine the correct digit of the image, we need to find a $Y = y_i$ such that $Pr(y_i|f_1, \ldots, f_m)$ is the maximum value among all the values in Y. This means we need to calculate $\arg\max\limits_y \; Pr(y|f_1, \ldots, f_m) = \arg\max\limits_y \frac{Pr(y)\prod_{i=1}^{m} Pr(f_i|y)}{Pr(f_1, \ldots, f_m)} \equiv \arg\max\limits_y Pr(y)\prod_{i=1}^{m} Pr(f_i|y)$ in this classifier.

Unlike the naive Bayes classifier, a perceptron does not use probabilities to make its decisions. It keeps a weight vector w^y of each class y. Given a feature list f, the perceptron compute the class y whose weight vector is most similar to the input vector f. This means given a feature vector f, we calculate $score(f, y) = \sum\limits_i f_i w_i^y$ and then we choose the class with the highest score as the predicted digit of the image. In order to calculate the weight vector in the learning process, we first need to calculate $y' = \arg\max\limits_{y''} score(f, y'')$. If $y = y'$, the instance is correct. Otherwise we need to make $w^y = w^y + f$ and $w^{y'} = w^{y'} - f$ until we figure out the correct result.

A MIRA classifier is very similar to a perceptron. But it uses $w^y = w^y + \tau f$ and $w^{y'} = w^{y'} - \tau f$ when $y \neq y'$. In MIRA, $\tau = min\left\{C, \frac{(w^{y'} - w^y)f + 1}{2|f|^2}\right\}$ where C is a positive constant. In this assignment, we will set $C = 0.001$ according to the recommendation.

The group members are Xiaohui Chen, Sarah Meger and Brian Monsen. We will demonstrate our result in class using slides and will submit the code to the TA.