Homework Assignment 3

Lecturer: Kyunghyun Cho

March 21, 2017

1. Suppose we are given a set of input-output pairs $D = \{(\mathbf{x}_1, y_1^*), \ldots, (\mathbf{x}_N, y_N^*)\}$, and we want to find the best classifier among the following hypothesis sets:

1. $H_{\text{perceptron}}$: Perceptron classifier
2. H_{logreg}: Logistic regression
3. $H_{\text{svm},C}$: Support vector machine with a regularization coefficient C

where C is one of $\{c_0, \ldots, c_M\}$. The dataset D is split into 80% training set and 20% test set. Our goal is two folds; (1) choose the best classifier, and (2) report its generalization performance (or its estimate). Describe in words how these two goals are met using the principles of K-fold cross validation.

SOLUTION: K-fold cross validation first splits the training set K times, giving us several training and validation sets. Each hypothesis set is then trained K times and the corresponding validation sets are used to find the empirical cost of each machine. For each hypothesis set, the collective empirical costs are averaged and compared amongst the various hypothesis sets. By training each hypothesis set multiple times on data that is statistically similar, we ensure that any outliers in the data are accounted for, intuitively giving us the best classifier. Getting the generalization is a trivial task. At the very beginning, 20% of the data was set aside for testing purposes and it was not used nor modified at all during this process. This gives us a reliable set of data to test our classifier with and to report the generalization error without much extra effort.

2. The distance function of multi-class logistic regression was defined as

$$
\begin{aligned}
D(y^*, M, \mathbf{x}) &= -\log p_{M^*(\mathbf{x})} \\
&= -a_{y^*} + \log \sum_{k=1}^{K} \exp(a_k),
\end{aligned}
$$

where

$$\mathbf{a} = \mathbf{W}\tilde{\mathbf{x}}.$$

Derive a learning rule step-by-step for each column vector $\mathbf{w}_c$ of the weight matrix $\mathbf{W}$.

SOLUTION: First of all, we know that $p(C=n|\mathbf{x}) = \frac{\exp(\mathbf{w}_n^\top \tilde{\mathbf{x}})}{\sum_{k=1}^K \exp(a_k)}$ for all $n \in \{1,\dots,K\}$
For all $y \in \{1,\dots,K\} \setminus \{y^*\}$
$\frac{\partial D(y^*,M,\mathbf{x})}{\partial \mathbf{w}_y} = -\frac{\partial}{\partial \mathbf{w}_y}\left(\mathbf{w}_y^\top \tilde{\mathbf{x}} - \log \sum_{k=1}^K \exp(a_k)\right)$
$= -(0 - \frac{\tilde{\mathbf{x}}\exp(\mathbf{w}_y^\top \tilde{\mathbf{x}})}{\sum_{k=1}^K \exp(a_k)}) = -(0 - p(C=y|\mathbf{x}))\tilde{\mathbf{x}}.$

For the y^*-th row vector
$\frac{\partial D(y^*,M,\mathbf{x})}{\partial \mathbf{w}_{y^*}} = -\frac{\partial}{\partial \mathbf{w}_{y^*}}\left(\mathbf{w}_{y^*}^\top \tilde{\mathbf{x}} - \log \sum_{k=1}^K \exp(a_k)\right)$
$= -(1 - \frac{\tilde{\mathbf{x}}\exp(\mathbf{w}_{y^*}^\top \tilde{\mathbf{x}})}{\sum_{k=1}^K \exp(a_k)}) = -(1 - p(C=y^*|\mathbf{x}))\tilde{\mathbf{x}}.$

In order to combine both equations, we need to replace certain notations. Each equation has a constant that is either 0 or 1, which merely corresponds to the desired output for the given input. In our combined equation, we can simply replace this constant with $\mathbf{y}^*$, which is a vector representing our desired outputs for all given input. When combining $p(C=y^*|\mathbf{x})$ and $p(C=y|\mathbf{x})$ for all $y \in \{1,\dots,K\}$, we get the probabilities of every possible y, thus being equivalent to our vector $\mathbf{p}$, which is the output of our machine. Substituting these values, we get $-\mathbf{x}(y^* - \mathbf{p})\top$. Our learning rule then becomes $\mathbf{w}_c \leftarrow \mathbf{w}_c - \mathbf{x}(y^* - \mathbf{p})\top$

3. PROGRAMMING ASSIGNMENT