Multiclass Classifiers

One vs All and All Pairs

Angela Zhu, Michael Lu, Qiming Fang, Jessica Wan

Math

- Representation

$$h_w(x)_j = \frac{e^{\langle w_j, x\rangle}}{\sum_{s=1}^{k} e^{\langle w_s, x\rangle}}$$

- Loss

$$L_s(h_w) = -\frac{1}{m}\sum_{i=1}^{m}\sum_{j=1}^{k}\mathbf{1}[y_i = j]\log h_w(x_i)_j$$

- Optimizer
  - Stochastic gradient descent

$$w = w - \alpha \nabla L_s(h_w)$$

Stochastic Gradient Descent

input:
training examples X,Y of size S
step size α
batch size $b < S$

```
converge = False
while not converge:
    shuffle X,Y
    prev_epoch_loss = loss(X,Y)
    for each batch:
        L_w = zeros((num_classes, num_features + 1))
        for x,y in batch:
            probabilities = softmax(x)
            for class j in classes:
                h_wx = probabilities[j]
                if y == j: Lw[j] += (h_wx - 1)*x
                else: Lw[j] += (h_wx)*x
        w -= alpha * Lw / num_examples_in_batch
    curr_epoch_loss = loss(X,Y)
    converge = diff(curr_epoch_loss, prev_epoch_loss) < threshold
```

Algorithms - Pseudo Code

All Pairs

- Train one binary classifier for each class pair.
- Use the argmax of the average of the binary classifiers to predict.

input:
training set $S = (x_1, y_1), ..., (x_m, y_m)$
algorithm for binary classification A
foreach $i, j \in \mathcal{Y}, i < j$:
$S_{i,j}$ initialize to be empty
for t=1,...,m:
if $y_t = i$, add $(x_t, 1)$ to $S_{i,j}$
if $y_t = j$, add $(x_t, -1)$ to $S_{i,j}$ let $h_{i,j} = A(S_{i,j})$
output:
the multi-class hypothesis defined by $h(x) \in argmax_{i \in \mathcal{Y}}(\sum_{j \in \mathcal{Y}} sign(j - i)h_{i,j}(x))$

One-vs-All

- Train one binary classifier for each class which predicts whether or not a sample belongs to that class (1) or not (0).
- Select the result of the classifier with the largest positive prediction.

input:
training set $S = (x_1, y_1), ..., (x_m, y_m)$
algorithm for binary classification model
foreach $i \in \mathcal{Y}$
let $S_i = \left((x_1, (1)^{\mathbb{1}[y_1 \neq i]}), ..., (x_m, (1)^{\mathbb{1}[y_m \neq i]})\right)$
let $h_i = A(S_i)$
output:
the multi-class hypothesis defined by $h(x) \in \text{argmax}_{i \in \mathcal{Y}} h_i(x)$

Previous Work

Iris Data + German Numerical Credit Data

- Combinations of our model + our LR, our model + SK LR, and SK Model + SK LR
- Iris Data:
  - 120 training size, 30 testing size
  - 4 features + 1 bias
  - 3 classes
- Credit Data:
  - 350 training size, 150 testing size
  - 69 features + 1 bias
  - 2 classes

Previous Work

Iris Data + German Numerical Credit Data

Iris Data

	All Pairs	One V All
SKLearn	**0.9**	**0.833**
Our Model	**0.933**	**0.966**

Credit Data

	All Pairs	One V All
SKLearn	**0.7028**	**0.7028**
Our Model	**0.7028**	**0.7028**

Summary

- Checked that our One vs All and All Pairs algorithms were correctly implemented no matter what our underlying binary classification algorithm A was
  - German credit data: had two classes
    - Our own One vs All and All Pairs had same predictions and accuracy with our own LogisticRegression binary classification algo, and with sklearn’s binary classification LogisticRegression algo
  - Unit tests:
    - Used sklearn’s OneVsRestClassifier and OneVsOneClassifier with underlying binary classification sklearn’s LogisticRegression algo against our own multiclass classifiers with sklearn’s LogisticRegression as the underlying binary classification
- Our own LogisticRegression and sklearn’s LogisticRegression algo are implemented differently, however:
  - Iris data: both our models One vs All and All Pairs with our own LogisticRegression and sklearn’s OneVsRest and OneVsOne classifiers with sklearn’s LogisticRegression give similar accuracies across multiple runs

Reflection

- Challenges:
  - Checking to see if our LogisticRegression for binary classification is accurate
    - Sklearn's LogisticRegression default uses L-BFGS optimization algorithm to find minimum of a function, other optimization algorithms available would be Stochastic Average Gradient, which would still be different from our implementation of Stochastic Gradient Descent
  - Debugging: for two classes One vs All and All Pairs should be the same
- Interesting to see how different optimization algorithms for the underlying binary classification algorithm would affect accuracy levels for One vs All and All Pairs multiclass algorithms
  - On Iris dataset, if we train and test on a random 80/20 portion of the dataset from shuffling, across multiple runs sometimes SGD is better sometimes L-BFGS is better