CSCE 636
HW1 Solution Keys

Yi Liu

10/2/2019

1. Data Preprocessing

- (a). *train_valid_split* : The function divides the raw training dataset into two sub-sets, training set and validation set, with a certain rate. The validation set is used to estimate the out-of-sample error of a trained model, to help us tune hyper-parameters and choose the best model

- (b). Yes. Firstly, more data is better. Secondly, validation set is originally part of the training set. When retrain the model before do testing, the whole training data is exposed, while the test data still keeps unseen.

- (d). We augment $\boldsymbol{x}$ by adding $x_0 = 1$ corresponding to the bias w_0. We can then write to the linear combination $\boldsymbol{w}^T \boldsymbol{x}$ for simplicity.

$$\sum_{i=1}^{d} w_i x_i + w_0$$

$$= \sum_{i=1}^{d} w_i x_i + w_0 \cdot 1$$

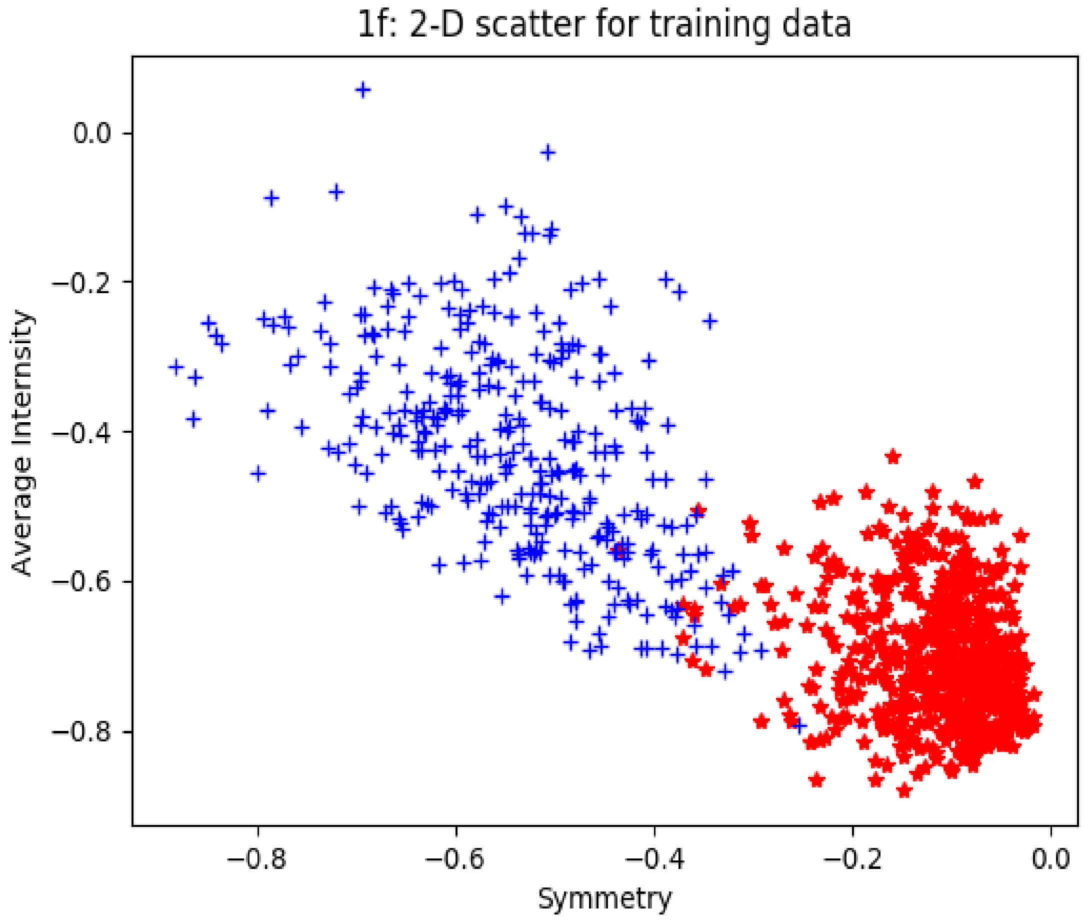
$\boldsymbol{w}$=$[w_0, w_1, \dots w_d]^T$

$\boldsymbol{x}$=$[1, x_1, \dots x_d]^T$

$$= \sum_{i=0}^{d} w_i x_i$$

$$= \boldsymbol{w}^T \boldsymbol{x}$$

- (f)


1f: 2-D scatter for training data
Average Internsity
Symmetry
0.0
−0.2
−0.4
−0.6
−0.8
−0.8
−0.6
−0.4
−0.2
0.0

2. Cross-entropy loss for logistic regression

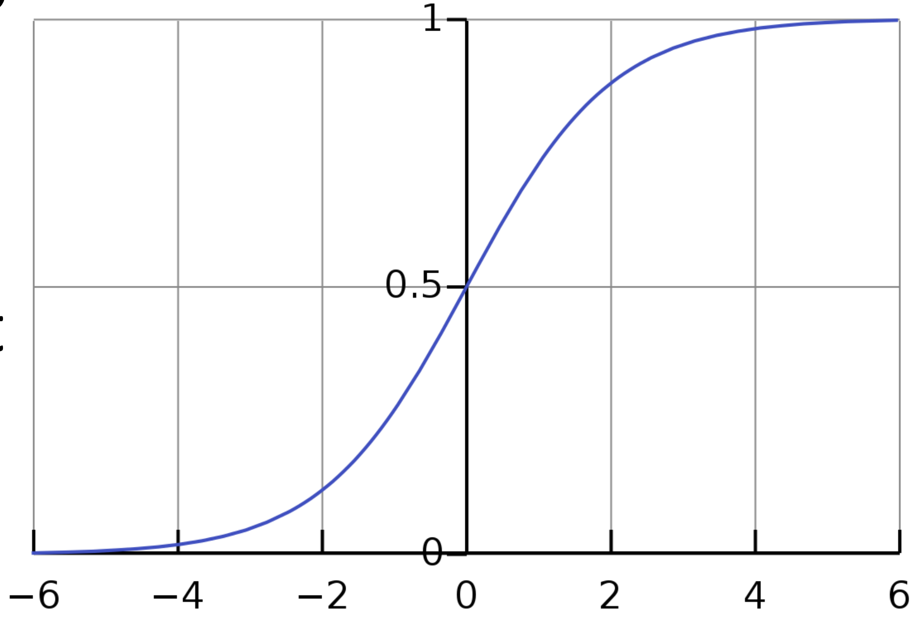
- (a,b)

$$E(\boldsymbol{w})_{(\boldsymbol{x},y)} = \ln(1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}})$$

$$\nabla E(\boldsymbol{w})_{(\boldsymbol{x},y)}$$
$$= \frac{1}{1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}} \nabla \left(1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}\right)$$
$$= \frac{1}{1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}} e^{-y\boldsymbol{w}^T\boldsymbol{x}} \nabla(-y\boldsymbol{w}^T\boldsymbol{x})$$
$$= \frac{1}{1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}} e^{-y\boldsymbol{w}^T\boldsymbol{x}} \nabla(-y\boldsymbol{w}^T\boldsymbol{x})$$
$$= -y\boldsymbol{x} \frac{e^{-y\boldsymbol{w}^T\boldsymbol{x}}}{1 + e^{-y\boldsymbol{w}^T\boldsymbol{x}}}$$

- (c). We can remove sigmoid function predictions, then the decision boundary is simply to be $\boldsymbol{w}^T \boldsymbol{x} = 0$. The sigmoid function is monotonically increasing function, making the decision boundary linear. We need to use the sigmoid function if we need to predict probabilities.

- (d). Yes, it's still linear. The sigmoid function is monotonically increasing function, and we just change the decision boundary to be $\boldsymbol{w}^T \boldsymbol{x} = \theta^{-1}(0.9)$.



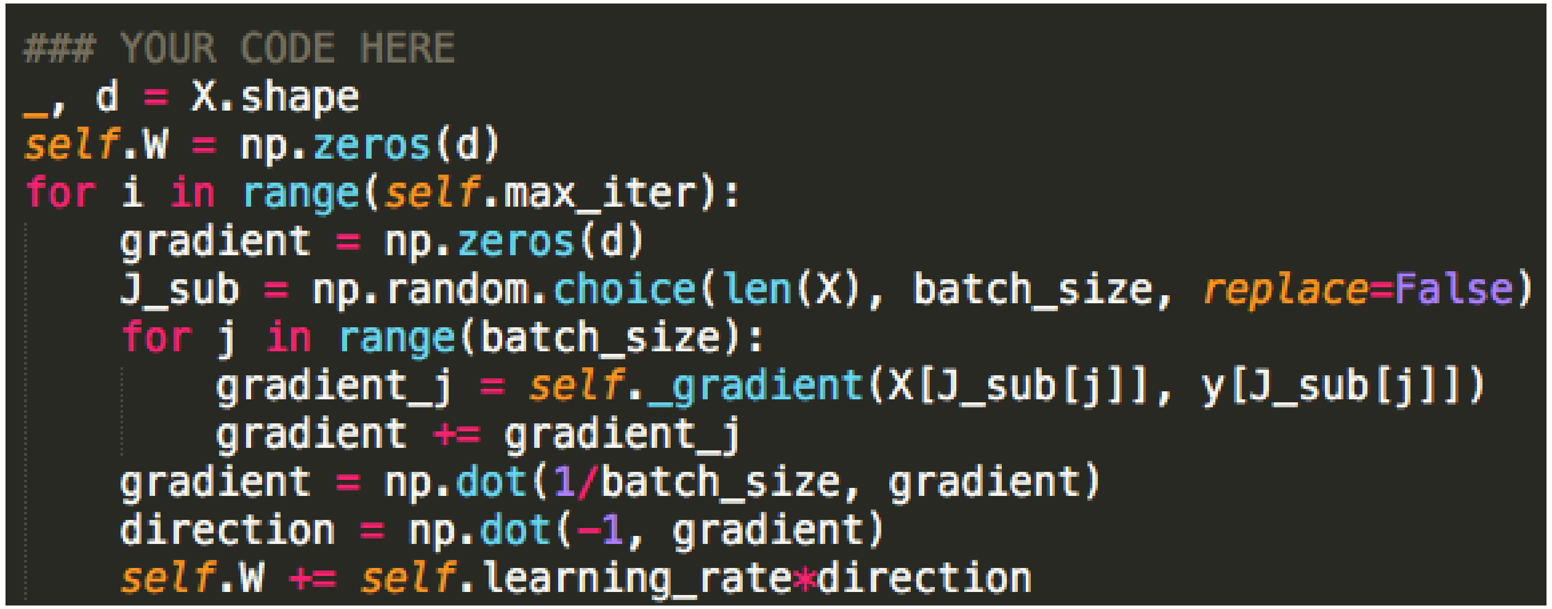
- (e). The used sigmoid function is monotonically increasing function.

3. Sigmoid logistic regression

- (a). $\nabla E(\boldsymbol{w})_{(\boldsymbol{x},y)} = -y\boldsymbol{x}\dfrac{e^{-y\boldsymbol{w}^T\boldsymbol{x}}}{1+e^{-y\boldsymbol{w}^T\boldsymbol{x}}}$
- (b). GD: One iter, update the whole data.

  SGD: One iter, randomly choose one sample to update weights.

  BGD: One iter, ‘randomly’ choose one batch of data to update weights.



```
### YOUR CODE HERE
_, d = X.shape
self.W = np.zeros(d)
for i in range(self.max_iter):
    gradient = np.zeros(d)
    J_sub = np.random.choice(len(X), batch_size, replace=False)
    for j in range(batch_size):
        gradient_j = self._gradient(X[J_sub[j]], y[J_sub[j]])
        gradient += gradient_j
    gradient = np.dot(1/batch_size, gradient)
    direction = np.dot(-1, gradient)
    self.W += self.learning_rate*direction
```

4. Softmax logistic regression

- (a). Check shape. Gradient matrix always has the same shape as weight matrix.

softmax + cross entropy

$$p_k = softmax_k(\boldsymbol{a}), \qquad L = -\sum_k y_k \log(p_k)$$

$$\frac{\partial L}{\partial a_k} = p_k - y_k, \qquad \frac{\partial L}{\partial \boldsymbol{a}} = [p_1 - y_1, \ldots, p_K - y_K] \epsilon R^{1\times K}$$

$$\boldsymbol{a} = XW, \qquad X \epsilon R^{1\times D}, \qquad W \epsilon R^{D\times K}$$

$$\frac{\partial L}{\partial W} = \frac{\partial L}{\partial \boldsymbol{a}} \frac{\partial \boldsymbol{a}}{\partial W} = X^T \frac{\partial L}{\partial \boldsymbol{a}} \epsilon R^{D\times K}$$

5. Sigmoid logistic vs Softmax logistic

(a). Binary LR: sigmoid + cross entropy

Multi-class LR: softmax + cross entropy

When K = 2 , the softmax-based multi-class LR is equivalent to the sigmoid-based binary LR.

(b). After training for one step, we can observe that

$$\nabla w = -\nabla w_1 = \nabla w_2.$$

Assume the learning rate of logistic regression and softmax classier are α_1 and α_2 separately. For the case

$$w - \alpha_1 \nabla w = w_2 - w_1 - 2\alpha_2 \nabla w,$$

If we want to ensure $w = w_2 - w_1,\ \alpha_1 = 2\alpha_2.$

That is, learning rate of the sigmoid LR is two times of that in softmax LR.