Robustness and fairness

Prof. Changho Suh, TA Jaewoong Cho, Gyeongjo Hwang

1. A setting where one can see a performance degradation due to poisoned data when applying non-robust algorithms (like logistic regression);
2. Another setting where one can see a fairness issue;
3. Learning-to-reweight algorithm for robustness;
4. Jafar's algorithm for fairness.

1. Data poisoning and robust machine learning

실습목표

-- Clean dataset과 Poisoned dataset을 인위적으로 생성한다.

-- 각각의 dataset에 logistic regression을 적용한 뒤 성능을 측정하고, poisoning 정도에 따라 성능 저하를 확인한다.

-- Poisoning data에 robust한 Learn to Reweight 알고리즘을 구현한다.

-- Poisoned dataset에 Learn to Reweight 기반 logistic regression을 적용한 뒤 성능을 측정하고, poisoning 정도에 따라 성능 저하를 확인한다.

1-A. Clean dataset과 Poisoned dataset을 인위적으로 생성

본 실습에서는 지난 Convex Optimziation for Machine Learning 실습시간에 구현한 모듈을 필요로 한다. 이를 불러오시오.

In [3]:
```
from blah import logistic_regression, generate_normal, measure_acc, etc
```

```
---------------------------------------------------------------------------
ImportError                               Traceback (most recent call last)
<ipython-input-3-1589cc285ac4> in <module>()
----> 1 from blah import logistic_regression, generate_normal, measure_acc, e
tc

ImportError: No module named blah
```

Clean dataset을 생성하시오

In [4]:
```
(X_clean_train, Y_clean_train), (X_clean_test, Y_clean_test) = generate_normal
(800, seed=2019)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-4-40df907f97b2> in <module>()
----> 1 (X_clean_train, Y_clean_train), (X_clean_test, Y_clean_test) = genera
te_normal(800, seed=2019)

NameError: name 'generate_normal' is not defined
```

dataset과 숫자가 주어졌을 때 해당 갯수만큼 label을 flip하는 함수를 구현하시오.

In [5]:
```
def simple_poison(X_clean_train, Y_clean_train, n_of_poison):
    '''
    Poison "n_of_poison" number of data points by flipping their labels

    Returns:
    X_poison_train, Y_poison_train
    '''

    n_of_poison = min(X_clean_train.shape[0], n_of_poison) # flip everything i
f n_of_poison > # of data pts

    ### START CODE HERE ### (≈ 4 line of code)
    X_poison_train, Y_poison_train = np.copy(X_clean_train), np.copy(Y_clean_t
rain)
    Y_poison_train[:n_of_poison] = 1 - Y_poison_train[:n_of_poison]
    ### END CODE HERE ###

    return X_poison_train, Y_poison_train
```

1-B. 불러온 logistic regression을 적용하여 성능을 측정

Poison rate을 아래와 같이 변화시키며 측정한다.

In [7]:
```
acc_poison = []
poison_rate = np.linspace(0, 1, 21)
### START CODE HERE ### (≈ 4 line of code)
for each_poison_rate in poison_rate:
    each_accuracy = Logistic(X_clean_train, Y_clean_test, X_clean_test, Y_clea
n_test)
    acc_poison.append(each_poison_rate, each_accuracy)
### END CODE HERE ### (≈ 4 line of code)
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-7-8d52d6669064> in <module>()
      1 acc_poison = []
----> 2 poison_rate = np.linspace(0, 1, 21)
      3 ### START CODE HERE ### (≈ 4 line of code)
      4 for each_poison_rate in poison_rate:
      5     each_accuracy = Logistic(X_clean_train, Y_clean_test, X_clean_tes
t, Y_clean_test)

NameError: name 'np' is not defined
```

그린다.

In [8]:
```
plt.scatter(poison_rate)
plt.xlabel('poison rate')
plt.ylabel('acc')
```

```
---------------------------------------------------------------------------
NameError                                 Traceback (most recent call last)
<ipython-input-8-e8c129a94191> in <module>()
----> 1 plt.scatter(poison_rate)
      2 plt.xlabel('poison rate')
      3 plt.ylabel('acc')

NameError: name 'plt' is not defined
```

1-C. Learn to reweight 알고리즘을 구현한다

Learn to reweight알고리즘은 ...

대충 수업에서 배운거 remind ...

저자 코드 최대한 활용하여 구현 따라하기

-clean validation셋이 필요하니까 위에서 데이터 생성할 때 train, val, test를 따로 받는 식으로 해도 괜찮을듯

-1) clean validation셋으로 loss define해서 back prop하는거 구현

-2) 계산된 weight을 기반으로 main model GD update하는거 구현

-3) 합치기

1-D. (1-B 반복)

2. Fair machine learning

실습목표

-- Fairness metric 측정하는 코드 구현

---diparate impact? disparate mistreatment? Jafar 논문 2개에 걸쳐 제안된 metric들

-- Fair dataset과 Unfair dataset을 인위적으로 생성한다.

-- 각각의 dataset에 logistic regression을 적용한 뒤 성능 및 Fairness를 측정

-- Jafar's fair learning 알고리즘을 구현한다. https://arxiv.org/pdf/1507.05259.pdf (https://arxiv.org/pdf/1507.05259.pdf)

--- Lambda가 주어지면 Loss + Lambda * Fairness-regularization-term을 GD로 푸는 알고리즘 구현

-- Unfair dataset에 Jafar 알고리즘 기반 logistic regression을 적용한 뒤 성능을 측정, lambda값에 따라 나온 (fairness & acc) tradeoff를 그린다