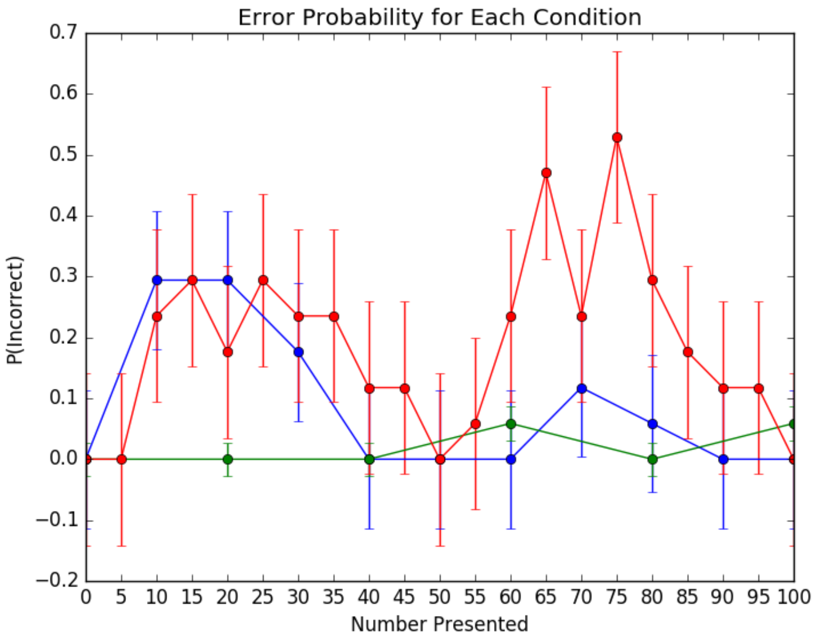
Jin Young Jeon
23051612
Cog Sci 190

Lab 4—Mutual Information

1.

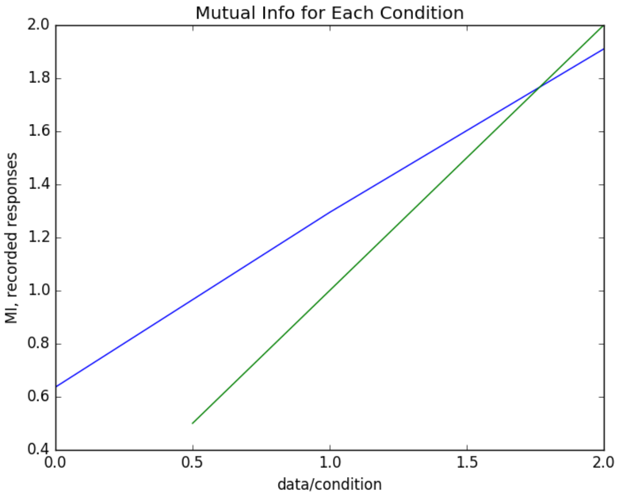


Green: Increments of 20 (Cond2) Blue: Increments of 10 (Cond1) Red: Increments of 5 (Cond3)

(For error probability, see lab4-mutualinfo.py
function error_prob computes individual error probability at given condition and trial)

2.

MI for condition 1:
1.29454516584
MI for condition 2:
0.636514168295
MI for condition 3:
1.91025216733



3. For some reason, my MI line crosses y=x. While the three MI points are supposed to take the curve form of log, my MIs appear almost linear. The more amount of data is shared, the more mutual information there is. Also, the more choices you have, ie. increments of 5 has more choices than increments of 20, the more randomness you have (the entropy increases).

(Appendix)

```
#1.
print ("data points")
data = np.genfromtxt('position-data.csv', dtype=int, delimiter=',', names=True)
print (data.tolist())
print ("\n")
print ("stimuli")
stimuli = np.genfromtxt('position-stimuli.csv', dtype=int, delimiter=',', names=True)
print (stimuli.tolist())

def error_prob(data, stimuli, condition, trial):
#computes error probability for specific trial that satisfy condition
    correct, incorrect = 0, 0
    if stimuli[trial][0] == condition:
        for j in range(0, len(data)):
            if stimuli[trial][1] == data[j][trial]:
                correct += 1
            else:
                incorrect += 1
    else:
        return "not conditionally met"
    return incorrect/(correct+incorrect)

plt.figure()
plt.errorbar(sorted_x1, sorted_cond1, xerr = None, yerr = 0.1131, marker = 'o')
plt.errorbar(sorted_x2, sorted_cond2, xerr = None, yerr = 0.0277, marker = 'o')
plt.errorbar(sorted_x3, sorted_cond3, xerr = None, yerr = 0.1413, marker = 'o')
plt.title("Error Probability for Each Condition")
plt.xlabel("Number Presented")
plt.ylabel("P(Incorrect)")
plt.xticks([0, 5, 10, 15, 20, 25, 30, 35, 40, 45, 50, 55, 60, 65, 70, 75, 80, 85, 90, 95, 100])
plt.show()

#2. Mutual Info
#MI of each individual condition from error probability
MI1 = sklearn.metrics.mutual_info_score(sorted_cond1, sorted_cond1)
print(MI1)
# 1.29454516584
MI2 = sklearn.metrics.mutual_info_score(sorted_cond2, sorted_cond2)
print(MI2)
# 0.636514168295
MI3 = sklearn.metrics.mutual_info_score(sorted_cond3, sorted_cond3)
print(MI3)
# 1.91025216733

x = ("Cond2", "Cond1", "Cond3")
y = (MI2, MI1, MI3)
plt.plot(y)
plt.plot([0.5, 1, 1.5, 2.0], [0.5, 1, 1.5, 2.0])
plt.title("Mutual Info for Each Condition")
plt.xlabel("data/condition")
plt.ylabel("MI, recorded responses")
plt.show()
```