Lab #4: Introducing Classification

Everything Data

CompSci 216 Spring 2015

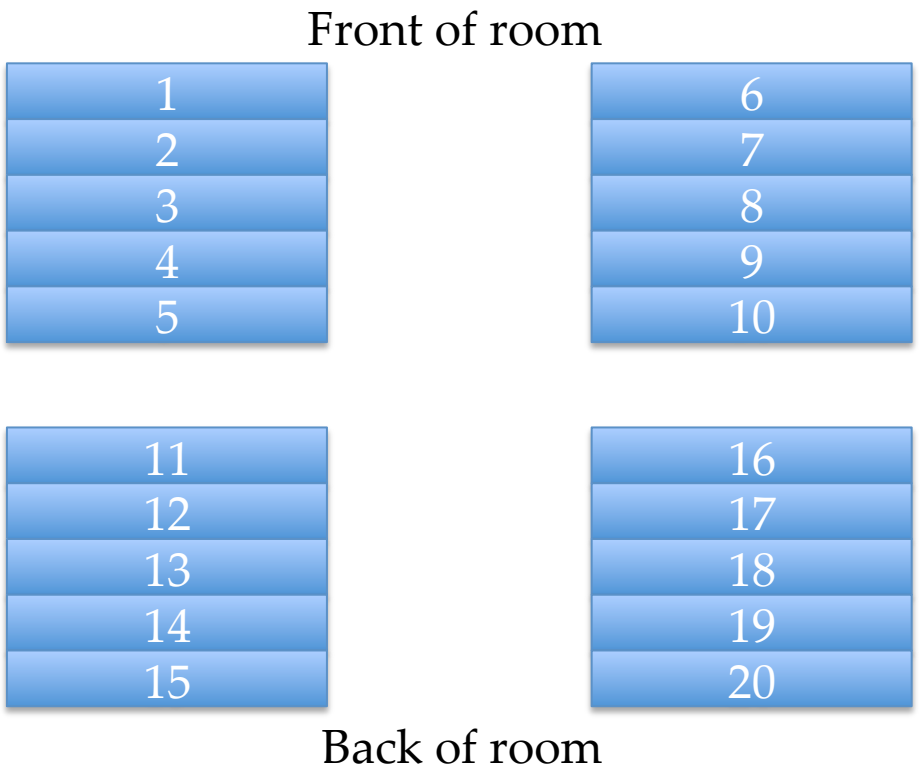
DUKE
COMPUTER SCIENCE

Announcements (Mon. Feb. 9)

- **New team assignment**
- **Project team formation** due in two weeks
  - 4 is the ideal team size; talk to us if you need special arrangement
- **Winner from Lab #3 challenge**:
  - (Old) Team 3:
    - Park, Grace
    - Jalan, Ishaan
    - Kuznetsov, Steve
    - Shaw, Scotty

Seat assignment

See course website (under “Schedule”) for team assignment



Format of this lab

- Discussion of Homework #4
- Introduction to classification
- Lab #4
  - Team challenge: win prize and extra credits!
- Discussion of Lab #4 (~10 minutes)

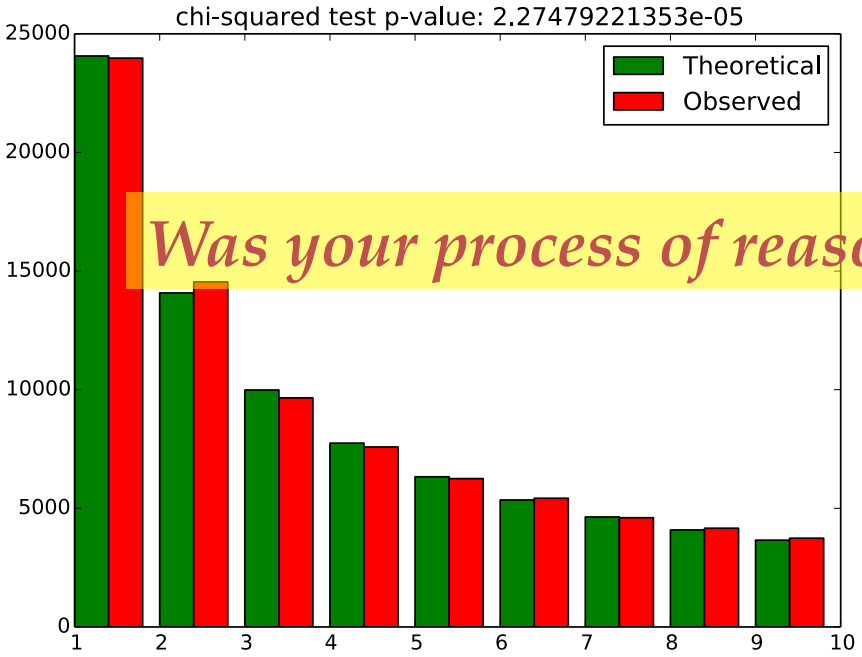
Homework #4, Part 1:

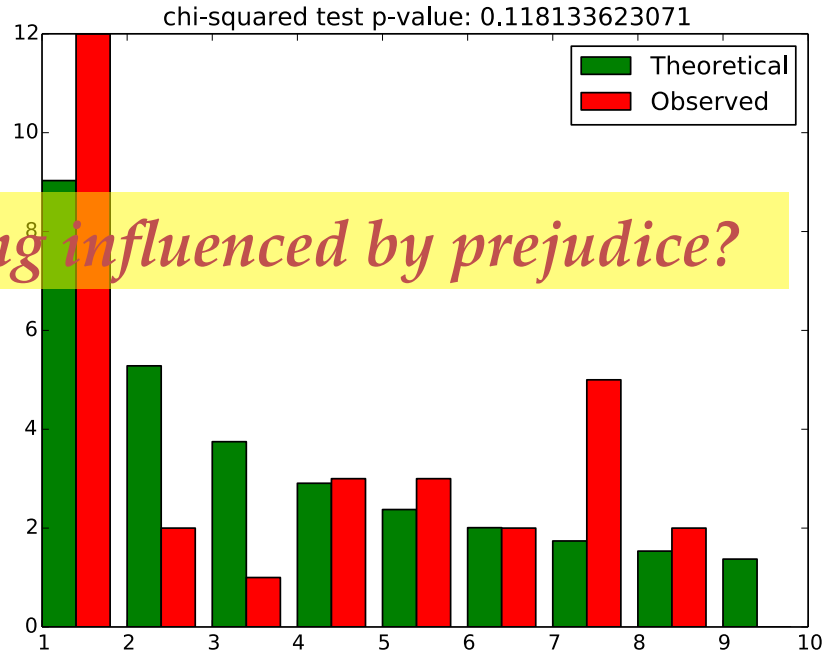
"Lucky" teammates

- Prob. of *A*, *B* in the same group again?
  - 3 (lucky slots) out of 63 (remaining slots)
  - About 5%
- Prob. that no pairs got lucky?
  - Rough estimate (with ***incorrect independence assumptions***): 6 chances to get lucky for each of the 16 groups
    - $(1 - 3/63)^{6\times16} \approx 0.9\%$
  - Simulation: ≈ 0.8%
  - More in sample solution

Homework #4, Part 2:

Benford's Law: population #'s





Was your process of reasoning influenced by prejudice?

Population numbers:
histogram similar,
but p-value small
because of large # of samples

Iranian election numbers (Rezai's):
histogram dissimilar,
but p-value not small
because of small # of samples

Homework #4, Part 3:

movielens data

(A) and (B) are straightforward

(C) Just looking at who rated A and B in data won't give you anything—***need to generalize*** what you see in data:

$$\begin{aligned}
&\mathrm{P}(U \text{ female} \mid U \text{ rates } AB) : \mathrm{P}(U \text{ male} \mid U \text{ rates } AB) \\
&= (\ \mathrm{P}(U \text{ rates } A \mid U \text{ female})\ \mathrm{P}(U \text{ rates } B \mid U \text{ female})\ \mathrm{P}(U \text{ female})\ ): \\
&\quad (\ \mathrm{P}(U \text{ rates } A \mid U \text{ male})\quad \mathrm{P}(U \text{ rates } B \mid U \text{ male})\quad \mathrm{P}(U \text{ male})\ ) \\
&\approx 0.11 : 1, \text{ which implies } \mathrm{P}(U \text{ female} \mid U \text{ rates } AB) \approx 0.10
\end{aligned}$$

- Compare with prior P(female):P(male) ≈ 0.41:1
- Basic idea behind ***Naïve Bayes Classifier***

Introducing Lab #5

Classification problem example: Given the set of movies a user rated, and the user's occupation, predict the user's gender

"Features"

m1	m2	m3	…	m1682	o1	o2	…	o21	gender
0	0	1	…	0	1	0	…	0	M
1	0	0	…	1	0	1	…	0	F
1	0	1	…	1	0	1	…	0	M
..	…	…	…	…	…	…	…	…	…
1	1	0	…	0	1	0	…	0	???
…	…	…	…	…	…	…	…	…	???

Categorical "outcome"

Each row is an "instance"

Training data
to teach your classifier

Test data
to evaluate your classifier

Accuracy = (# test records classified correctly) / (# test records)

Where is test data?

What if no test data is specified, or we don't know the right answers?

- We can still evaluate our classifier by splitting the data given to us

Rookie mistake:
train and test using the same (whole) dataset

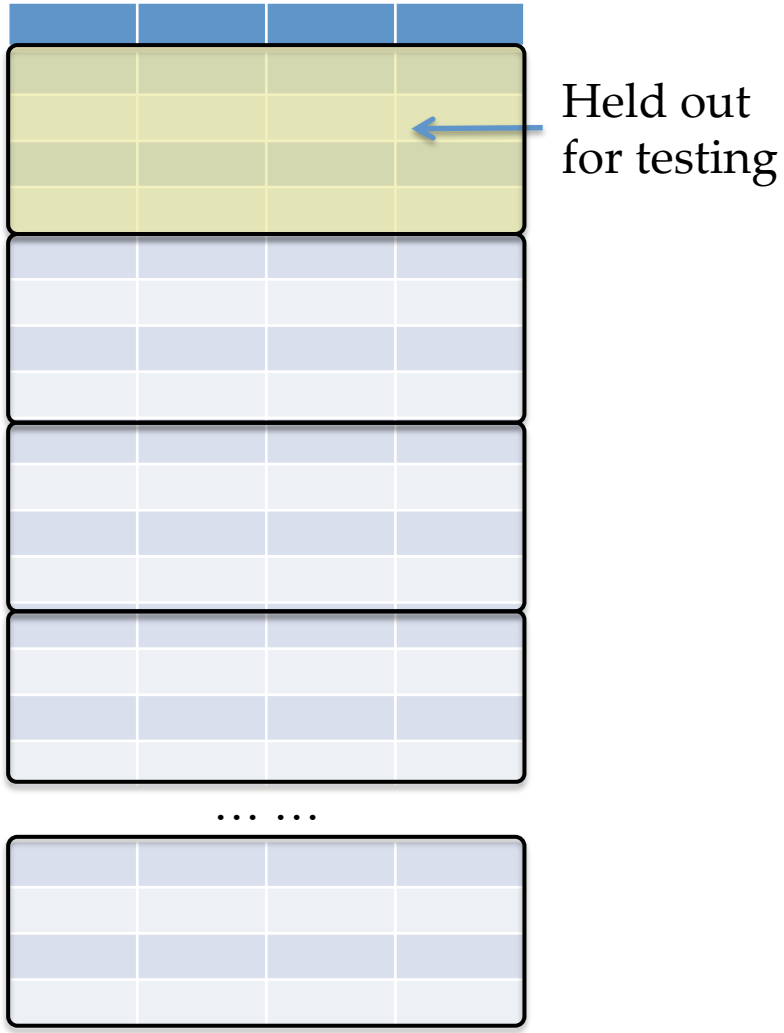
Lucky splits, unlucky splits

- What if a particular split gets lucky or unlucky?
- Should we tweak the heck out of our classification algorithm just for this split?

☞Answer: ***cross-validation***, a smart way to make best use of available data

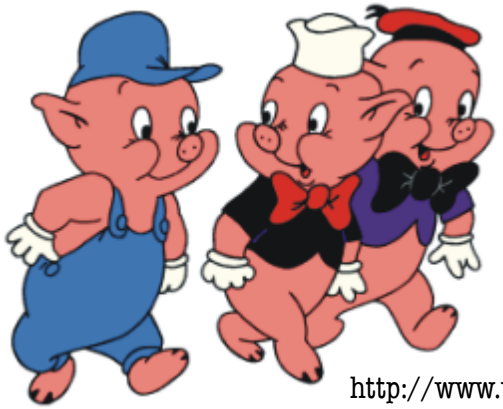
r-fold cross-validation

- Randomly divide data into *r* groups (say 10)
- Hold out each group for testing; train on the remaining *r* – 1 groups
  - *r* train-test runs and *r* accuracy measurements
  - A better picture of performance



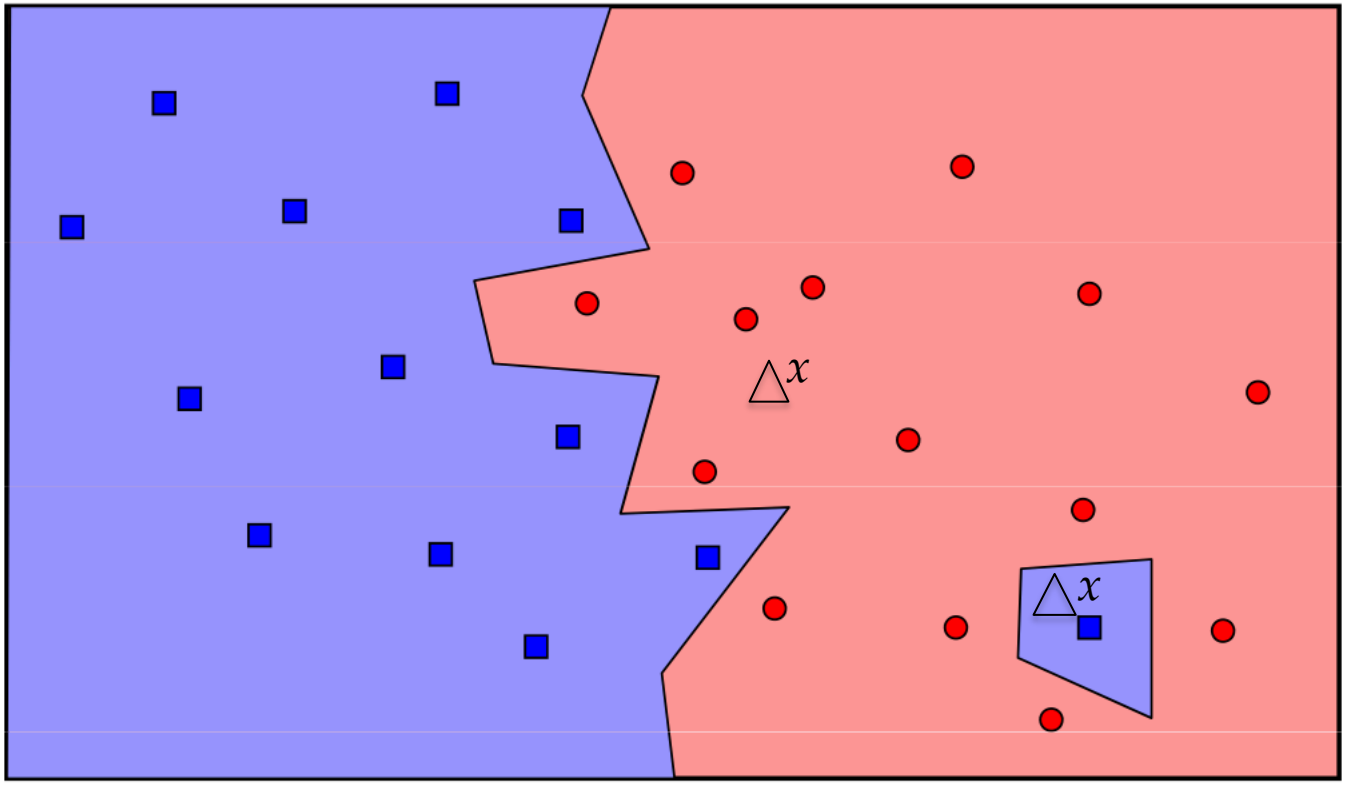
Three little classifiers

- `classifyA.py`: a "mystery" classifier
  - Read the code to see what it does
- `classifyB.py`: Naïve Bayes Classifier
  - Along the same line as Homework #4, 3(C)
- `classifyC.py`: *k*-Nearest-Neighbor Classifier
  - Given *x*, choose the *k* training data points closest to *x*; predict the majority class



http://www.weirdspace.dk/Disney/ThreeLittlePigs.htm

More on the *k*NN classifier



$k = 1$

Source: Daniel B. Neil's slides for 90-866 at CMU

Team work

1. Train-Test Runs and the Mystery of *A*
   (A) Which classifier seems to work best?
   (B) What exactly does *A* do?

Get checked off on these

2. Tweaking *k*NN
   (A) How does *k* affect accuracies on training vs. test data? Is big or small *k* better for this problem?
   (B) How does $k = 500$ compare with *A*?

Get checked off on these

Team challenge

The Evil SQL Splitters: find a train-test split such that the classifiers are great on training data but horrible on test

Redemption of Naïve Bayes: find a train-test split such that *B* beats *A* and *C* hands-down

- Extra credit worth 5% of a homework if 4× and *B* has ≥60% accuracy; must get checked off in class

Reward for best answers

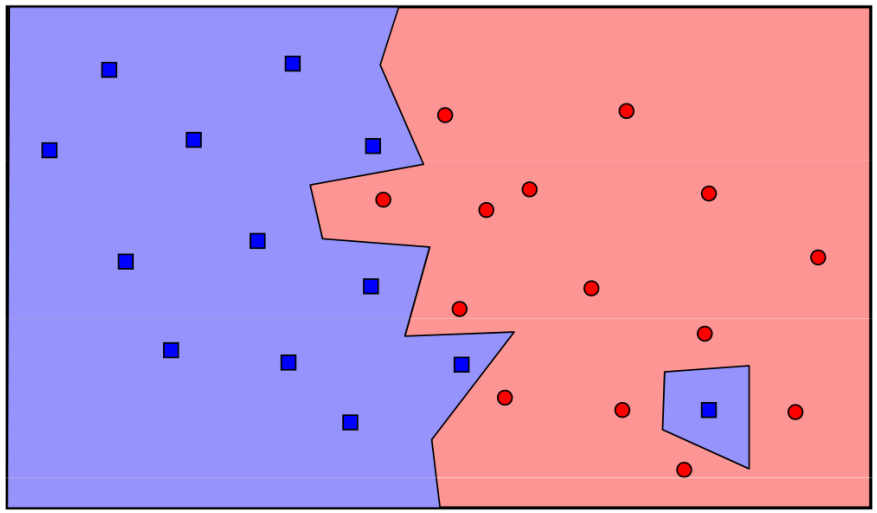
Discussion: Parts 1 & 2

1. Train-Test Runs and the Mystery of *A*
   (A) Which classifier seems to work best? *A*
   (B) What exactly does *A* do? *Just looks at M/F ratio in training data; doesn't even use other features*
2. Tweaking *k*NN
   (A) How does *k* affect accuracies on training vs. test data? Is big or small *k* better for this problem? *Training accuracy goes down, but test accuracy goes up; bigger k seems better here*
   (B) How does $k = 500$ compare with *A*?

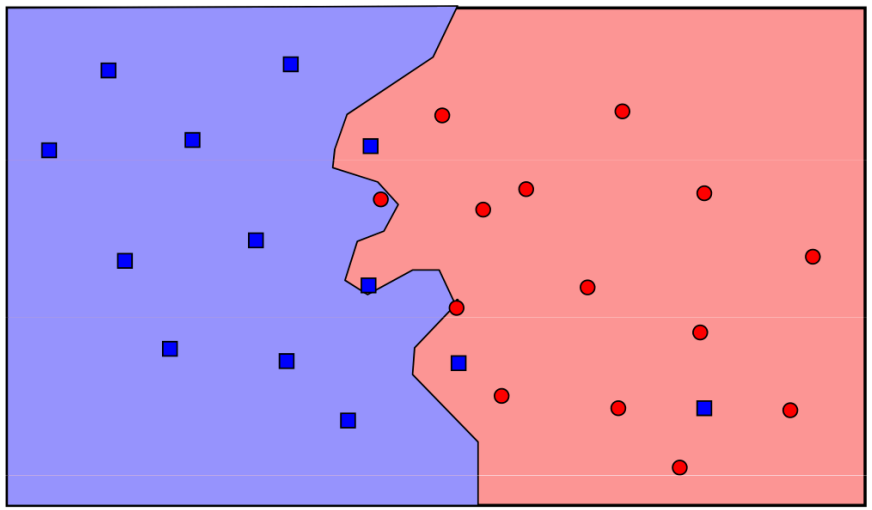
It approaches A—it basically goes by M/F ratio in a significant fraction of the training data

Lessons learned:

Overfitting hurts generalization



1NN: fits training data perfectly,
but handles noise and outliers poorly
and doesn't generalize well



3NN: has a smoother "boundary,"
and is less susceptible to noise
and outliers

What if we set k really, really big?
The opposite happens: ***underfitting***

Image source: Daniel B. Neil's slides for 90-866 at CMU

Discussion: Part 3 challenge

- *The Evil SQL Splitters:* find a train-test split such that the classifiers are great on training data but horrible on test
  - *Make training all males except one; test is all-female*
    - *B and C learn very little about females*
    - *A is messed up by the wrong ratio*

  Training accuracy: ~1
  Test accuracy: ~0

Redemption of Naïve Bayes: find a train-test split such that *B* beats *A* and *C* hands-down

- *Start with above split, but add some females (~70)*
  - *Not enough to sway the ratio to save A (0 accuracy)*
  - *Not enough to cover the space for C (0.12 accuracy)*
  - *B becomes better faster with more females (0.61 accuracy)*

Lessons learned:

Usefulness and limitation of CV

- Smart reuses of available data
  - Paints a broader picture of accuracy
  - Allows tuning of "hyper" parameters
    - E.g., *k* in *k*NN
- Again, don't test on data you train with
  - If you also need to tune, split data into train-validate-test
- Still, the "real test" remains unseen
  - No amount of cross-validation will help if your data collection is flawed

Finally

- Remember to **submit** `team.txt` **by midnight**
- Sample solutions to Lab #4 will be posted by tonight