Our project will involve ways to improve classifier performance where the training data is labeled by human raters.

Background: For example, suppose we want to train a classifier that inputs a web page and outputs whether the web page contains significant tobacco content. We train the classifier by example: we show it a bunch of pages where the correct label (tobacco or not-tobacco) is known. From these examples, the classifier learns how to label an unseen web page. In some machine learning problems, labels come automatically. For example, whether a person clicks on an ad or not is something we can log and we therefore get training data for free, to train a classifier that predicts whether a user will click. In other cases, which we consider here, there are no automatic labels. For example, we don't have a way of automatically determining whether a page has tobacco content. We instead sample a set of web pages and get humans to label them manually.

This work leads to a variety of problems:
1. Sometimes the humans make mistakes. With a finite budget for paying labelers, should we get each web page labeled multiple times and, for example, take the majority vote, or should we get more web pages labeled? Which leads to a better classifier?
2. Some humans are better labelers than others for a specific task. With a large pool of raters, how do we test and select which raters to send each web page to balancing time to get ratings (choosing only one rater will make the rating task take too long), cost of tests and ratings, and quality.
3. Some of the events we are looking for, like tobacco, occur very rarely. So uniformly sampling web pages will result in few positives. To get a sufficient number of positives, we'd need to label a prohibitive number of web pages. How do we deal with this problem in getting train and eval data. How can we eval in this context?
4. If we have already labeled 2000 web pages, and we have budget to label 3000 more, how do we choose the 3000 that will be most useful in improving the performance of the classifier?

And others.

We would like students to have a strong coding background, be proficient in statistics (confidence intervals, statistical significance, sampling, and probability), and have basic familiarity with machine learning. As always, the funding division loves it (and we do too) when we have strong female representation.

References?