

# Projects for ICS273A Fall '06

---

Max Welling,  
UC Irvine

## 1 Project 1

The goal of this project is to predict access-point (AP) traffic, i.e. the number of users being connected to  $AP_a$  at time interval  $t$  on day  $d$ . There are  $A$  AP points.

The data will consist of  $D$  days (about 100) worth of data. Each day is binned into  $T$  intervals (about 100), where at each time interval you will see a count value  $N_{t,a,d}$  which represents the number of users connected to AP  $a$ , at time  $t$  on day  $d$ .

We will also provide a test set, on which we have deleted all counts after a randomly chosen time  $t^*$  (different for each day). You should try to predict the number of users connected to each AP for the remainder of the day.

You are completely free to use your own imagination, but here are some hints. To predict AP counts for some time  $t$ , you can use the counts from a small window of time steps that preceded it as your features. Let's say you use  $h$  steps in the past. You can stack the counts for all the AP's for  $h$  time points into a long vector of length  $A \times h$ , and use these as features for your classifier. Or you can try to compute some distance between AP count traces and use these in a kernel method. So, you could create such a dataset by cutting the day into chunks of size  $h$ , where the AP counts at time  $h + 1$  acts as the label. You could learn different classifiers for different days of the week.

Once you have a classifier, you can imagine running it forward in time. For the first  $h$  time slots you simply use the average from the training data. After that you can run your classifier and predict the counts for  $h + 1$ , etc. forward in time. Clearly the predictions may be pretty good right after time  $t^*$ , but degrade pretty quickly.

Here is another approach. Try to cluster days or time windows into groups. Then, if you need to predict the AP counts of some time  $t$ , first find the appropriate cluster for that window (treating the last time point of that window as missing), and predict the AP count as an average of the last AP values over the windows in the cluster.

There are many more things you can do, so use your creativity. Only the final results count.

## 2 Project 2

Subscribe as a team to the netflix challenge: [www.netflixprize.com](http://www.netflixprize.com) Download the data, and perhaps split off a smaller subset of the data to work with.

You should submit predictions and report on your error that netflix computed for you.

A very useful reference is the masters thesis and papers of Ben Marlin at:

<http://www.cs.toronto.edu/~marlin/research/research.shtml>

You can organize the data as a large matrix over movies and users with ratings between 1..5. Note that most entries are missing! You can easily compute a similarity between users by computing the inner product between their video ratings as a long vector  $R_v(u)$  where  $v$  indexes the video and  $u$  the user. Note that this vector is very sparse (many 0's for unrated movies). In matlab you can represent vectors in sparse format and quickly compute an inner product.

For a new user, simply compute the inner product with all other users and retain the  $K$  nearest neighbors which also rated the movie you want to predict the rating for. You can now compute some weighted average (weighted by similarity) to predict the rating.

Again, there are tons of things you can try. So be inventive and become famous and rich (note, your chances are a lot better than in the lotto).