Project 1

The goal of this project is to explore different methods for converting images to vectors and then using this representation for doing recommendations

1. Data preparation

Download the imagenet collection from http://image-net.org/. The imagenet collection is very big for the scope of the class. We will pick few categories so that we can keep the computations tractable. As you can see http://image-net.org/explore the categories are not balanced. For the scope of this class we will use the images from:

- Plant, flora, plant life (1271)
- Geological formation (1808)
- Fungus (1207)
- Sport (1888)
- Person (1242)
- Animal (1571)

2. Converting images to vectors

In this part of the project you will have to convert the images to vectors. The following methods should be used:

1. Autoencoder (unsupervised): As presented in class. You are welcome to build your own in tensorFlow, here is an example, or you can use other implementations as in theano, caffe and keras.
2. Singular Value Decomposition: Represent each image as a vector by unfolding it columnwise. If you have N images, and each image is 256 x 256 that will give you a matrix of N x 2^16. You can do SVD on that matrix and transform each image as a k dimensional vector. You may choose to resample images to be uniform size beforehand.
3. Histograms: For every image:
    a. Create RGB histograms and combine to create a single feature vector by concatenating RGB vectors.
    b. Create HSV histograms and combine to create a single feature vector by concatenating HSV vectors.
4. Recommender system (see #3 below)

3. Converting images to vectors with recommender systems

In this part of the project you will simulate an image recommender system.

1. Assume that you have N=1000 users
2. Create an index number for every image starting from 0.

3. Simulate user ratings
    a. For every user create generate ratings for M randomly selected images. M ~ Geometric(0.1) for 4 out of 6 different categories.
    b. For every rating generate a score S ~ truncnorm(mu=category_mean, upper=5, lower=1, sigma=1.) The category mean should be distributed as uniform random variable between 1.5 and 4.5, cm ~ Unif(1.5, 4.5).
    c. As an example: user Nick might have 15 ratings for Plant, 55 ratings for fungus, 20 ratings for person, and 5 ratings for animal. The category means might be 2.0, 4.2, 3.0, 3.5. All individual ratings will range between 1 and 5.
4. Using the user-image ratings matrix do SVD and find a vector representation of the images.

4. Comparing different methods

Each of the four techniques in #2 above convert images into a vector. For each of the four techniques:

1. Use Euclidean distance and Pearson correlation coefficient to find the 5 nearest neighbors of each image.
2. Use majority vote of the 5 NNs to determine the category of each image.
3. Create a 6x6 confusion matrix for each distance metric whose rows index the true categories and whose columns index the predicted category values. Compute counts for each cell, and the overall accuracy of the system (percentage of correctly classified images).

4. Deliverables

It is recommended to train your models for different parameters, objectives. You should present a detailed report on your experiment that explains your design choices and results. Are the results reasonable? How did you test to assure so? Reports should also include code in python or ipython notebooks.