ECS271 – Machine Learning

Professor Ian Davidson

TA - Hongjing Zhang

Enrolled Students

Overview of First Lecture

- Our aim is to let you know what you are in for!
- Modules in course
- Style of each module
- Assessments (lots of details on assignments)
- Questions and Answers

Modules

- There are many types/settings in machine learning (ML) and many variations of each type.
- We will focus on five common ML settings.
  - Each is typically formulated as an optimization problem where we learn a function of some sort.
  - Each setting differs by what we are trying to learn to perform and what annotations/information we have to learn from.
- Please give me some examples of ML **settings** you know of?

Five x Two Week Modules in This Course

- Unsupervised learning
- Supervised learning
- Reinforcement learning
- Representation learning
- Outlier detection (typically unsupervised)

- (Typically Week 1) For each we will cover from a text (lots of chalk board stuff)
  - Problem setting and basic formulation
  - Derive classic algorithms from first principles
  - Derive variations
- (Typically Week 2) Then we will transition to deep learning formulations (you will help to present these)
  - Hongjing and I will do the first few to set the standard.
  - These will be in the form of papers/surveys
  - Will cover emerging topics such as fairness and explanation

Assessments

- Quizzes to test fundamental understanding of non-deep formulations from texts (20%)
- Written summaries to test understanding of DL papers (20%)
- 3 x 20% for each programming assignment
  - These are completed individually
  - Really lets you understand the material by trying them on a challenging problem
  - Open ended questions but you must use methods in the class
    - Some comparative assessment for bonus points.

1st Assignment – Unsupervised Learning

Setting

- The training data set consists of 1 million transactions/records (user-movie pairs) of features/attributes:
- <movie id, customer id, rating, date recommended>
  You may discretize/aggregate features such as date if you so wish.
- **If you have not used Netflix, all you need to know that after you view a movie, you are asked to rate it from 1 to 5 stars.**

Example Data

Training Data

head train.csv

```
movie-id,customer-id,rating,date
0001262,24294,4,2004-07-04
0001810,18187,3,2004-03-12
0002000,23573,5,2005-01-20
0003312,25907,1,2005-05-12
0003371,21487,5,2004-05-06
```

head test.csv

```
0011089,11765,?,2005-06-03
0011279,5858,?,2005-05-07
0012140,5202,?,2004-01-24
0012191,18818,?,2005-03-27
0012299,10021,?,2005-09-07
0012605,12449,?,2005-08-03
```

```
movie_id, year_produced, title
1,2003,Dinosaur Planet
2,2004,Isle of Man TT 2004 Review
3,1997,Character
4,1994,Paula Abdul's Get Up & Dance
5,2004,The Rise and Fall of ECW
6,1997,Sick
7,1992,8 Man
8,2004,What the #$*! Do We Know!?
9,1991,Class of Nuke 'Em High 2
```

The Tromaville Institute of Technology (T.I.T.), **located** inside the nuclear plant,

Professor Holt who has perfected a race of 'Sub-humanoids'; Living beings without emotions,

Roger Smith meets a beautiful subhumanoid named Victoria, they fall in love. …

Tsave Victoria from this messy fate, but first he'll have to face the giant mutant squirrel, Tromie,

You Can View The Data as an Incomplete Matrix (or Tensor)

Movies

Users

? ? ? ? ? 5 ...

? ? 4 ? 3 1 ...

3 3 2 2 3 4

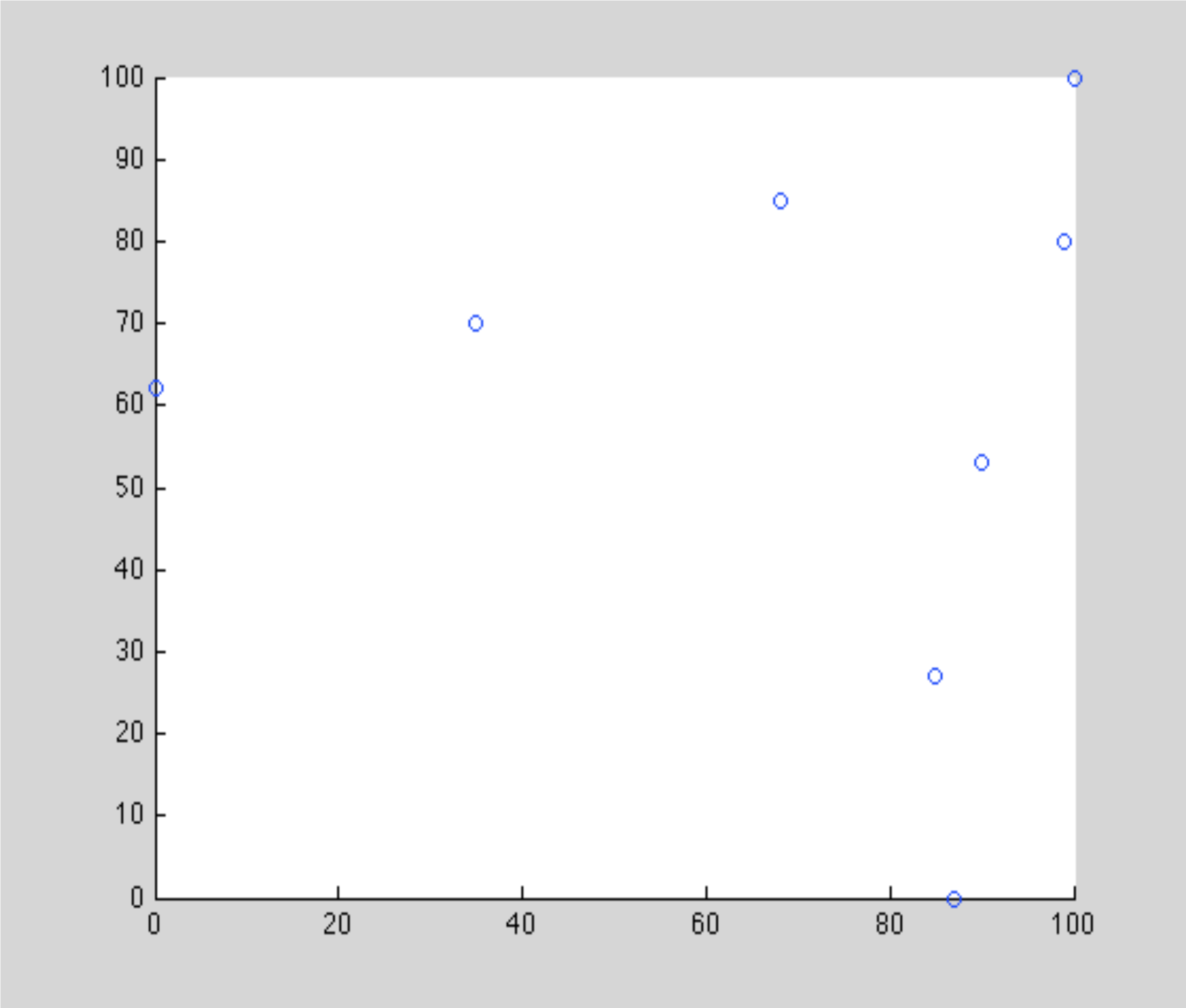
People's rating are varied even by density

Two Main Tasks (Scored Separately)

- Predict an integer rating for each of the 250,000 hold out <user,movie> pairs
- Task 1: Using spectral clustering methods
- Task 2: Using matrix completion methods

2nd Assignment – Learning to Predict Digits

0	62
35	70
68	85
100	100
99	80
90	53
85	27
87	0



2^{nd} Assignment

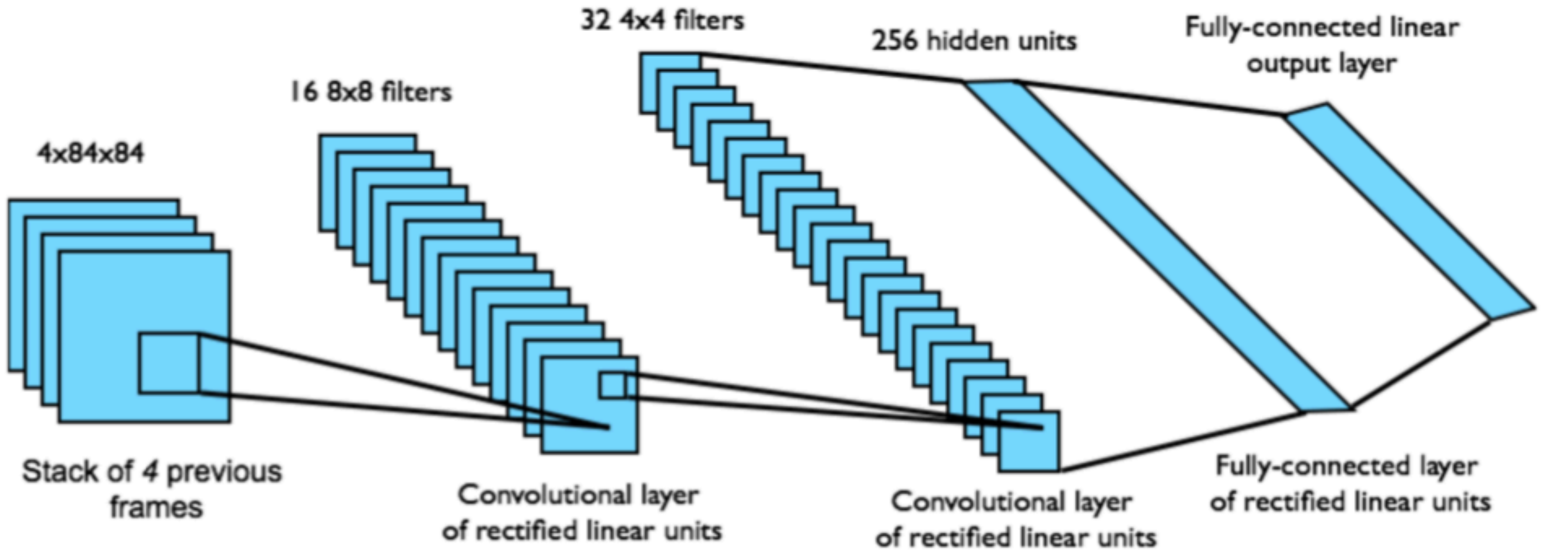
- In this assignment you will use a variety of support vector machines (SVMs) to predict hand written digits. These digits were written using a pen stylus by 40+ different authors.
- The x,y position of the pen was captured eight times hence x1,y1 is the position of the pen when it first touched the paper and x8,y8 is the last pen position. For ease of processing the co-ordinate system is 0-100 for both the x and y dimensions.
- Providing all 8 co-ordinate pairs makes the problem very easy so instead we will use positions (x3,y3) to (x6,y6).
- The training data is here studentspen-train.csv Download studentspen-train.csv

- You only need to use the primal form of SVMs for this assignment.
- Question 1): Train a linear multi-class classification SVM with no kernel. Specify i) Your mapping function and ii) Your loss function (20 points)
- Question 2) Describe a method to estimate your performance using an empirical method. Compare this estimate with a well known theoretical bound. Explain why/if there is a difference. (5 points)
- Question 3) Submit your predictions on this test set, one prediction per line in the order given studentsdigits-test.csv Download studentsdigits-test.csv (10 points)
- Question 4) Implement both types of transfer learning SVM (hypothesis and instance transfer) to train 1 vs 7 (target problem) by transferring in 1 vs 9 (source problem). Report your error estimate for the target problem with i) no transfer, ii) hypothesis transfer and iii) instance transfer. Which performs better? Why? (20 points)

3rd Assignment Overview Implementing the Deep Mind Paper To Play Atari 2600

Using the Gym OpenAI Platform

- See https://gym.openai.com/ and https://gym.openai.com/envs/#atari
- Two versions of every game based on state space: RAM vs Image. We'll use later.
- We will focus on Space Invaders and Convolutional neural networks



Code Description

- Two files
  - dqn.py (backend)
  - Run_dqn.py

Assignment

We provide some starter code here. You will need to have access to a GPU or use the CSIF machines.

Some of the CSIF machines in Kemper Hall now have GPU cards. See http://csifdocs.cs.ucdavis.edu In particular it seems Kemper 71 and Kemper 75 have about 20 GPU machines each.

The writeup for the assignment is here assignment3-1.pdf

Since this is the first time we've tried the assignment, there maybe updates of code and writeup requirements!

Finally the original DeepMind paper is here DeepMind-dqn (1).pdf

The assignment as of now contains no bonus points. I'll add bonus questions some shortly.

Questions for Assignment

2 Part 1: Extend the Deep Q-learner (60 points)

1. (**written**) Explain what is a replay memory/buffer in DQN. Why it is necessary?
2. (**coding**) Implement the "randomly sampling" function of replay memory/buffer (see line 89 of `dqn.py` for TODO and hints). Submit necessary code as sampling.zip.
3. (**coding**) Given a state, write code to calculate the Q value and the corresponding chosen action based on neural network (see lines 50 $\sim$ 55 in `dqn.py`). Submit necessary code as action.zip.
4. (**written**) Given a state, what is the goal of line 48 and line 57 of `dqn.py`? Aren't we calculating the Q value and the corresponding chosen state from the neural network?

Questions for Assignment

5. (**coding**) Implement the "Temporal Difference Loss": the objective function of DQN (see lines $69 \sim 73$ of `dqn.py`), which is described in the Mitchell Q-learning text on the website:

$$y_i = \mathbb{E}_{s' \sim \mathcal{E}}\left[r + \gamma \max_{a'} Q(s', a; \theta_{i-1}) \middle| s, a\right] \tag{1}$$

$$\mathcal{L}_i(\theta_i) = \mathbb{E}\left[(y_i - Q(s, a; \theta_i))^2\right] \tag{2}$$

   Submit necessary code as TD.zip.

6. (**written**) After you make these changes, train DQN by running `run_dqn_pong.py`. To achieve relatively good performance, you need to train more than 1000000 frames. It takes $\sim$ 10 hours on a Google Cloud Virtual Machine with 2 vCPUs and 1 NVIDIA Tesla K80 GPU.

   Explain what parameter tuning you performed such as:

   a. You might tune the hyper-parameters like γ and initial size and the size of replay buffer to gain a better performance.

   b. Modify `run_dqn_pong.py` to track and plot how loss and reward changes during the training process. Attach these figures in your report. Your grade will be based on the ranking of final reward.

Questions for Assignment

Given a trained DQN, we can evaluate the DQN on the input frames and activations of the last layer of the neural network. For example, if we evaluate the trained DQN on X frames and dimension of the last hidden layer has Y nodes, we are able to collect data of form $X \times Y$. Here you will use one or more of the dimension reduction techniques covered in class (PCA, CCA, MDS, ISOMAP or LLE) To visualize the (manifold or otherwise) structure of those frames/activations by mapping them to a 2-D space.

This embedding could be useful to analyze DQN and somewhat explain why DQN can achieve good performance. Here is an example: If we record the chosen actions of all "exploitation" frames, we will be able to see whether similar outputs of neural network lead to the same chosen action. The side information (*e.g.* screenshot of game board, reward, chosen action, state of the board, order of the frames, etc) that could be recorded when evaluating trained DQN on frames are very fruitful and could explain different aspects of the trained Q-learner.

a. Evaluate your trained DQN in Part 1 on (*e.g* randomly picked 1000) frames and collect the features as well as various side information.

b. Perform reduction on the features.

c. Analyze the embedding based on two kinds of your collected side information.