The Coders' Club

Machine Learning: G1

Week 9: Assignment

Topics:
Anomaly Detection
Recommender Systems

Some Additional Courses:

- Machine Learning Onramp (MathWorks)
  https://www.mathworks.com/learn/tutorials/machine-learning-onramp.html

- Deep Learning Onramp (MathWorks)
  https://www.mathworks.com/learn/tutorials/deep-learning-onramp.html

- AI From the Data Center to the Edge – An Optimized Path Using Intel® Architecture (Intel AI)
  https://software.intel.com/en-us/ai/courses/data-center-to-edge

- Machine Learning (Intel)
  https://software.intel.com/en-us/ai/courses/machine-learning

- Deep Learning (Intel)
  https://software.intel.com/en-us/ai/courses/deep-learning

Anomaly Detection

Q.1. For which of the following problems would anomaly detection be a suitable algorithm?

- Given a dataset of credit card transactions, identify unusual transactions to flag them as possibly fraudulent.
- Given an image of a face, determine whether or not it is the face of a particular famous individual.
- From a large set of primary care patient records, identify individuals who might have unusual health conditions.
- Given data from credit card transactions, classify each transaction according to type of purchase (for example: food, transportation, clothing,)

Q.2. Suppose you have trained an anomaly detection system that flags anomalies when $p(x)$ is less than ε, and you find on the cross-validation set that it has too many false negatives (failing to flag a lot of anomalies). What should you do?

- Increase ε
- Decrease ε

Q.3. Suppose you are developing an anomaly detection system to catch manufacturing defects in airplane engines. Your model uses

$$p(x) = \prod_{j=1}^{n} p(x_j; \mu_j, \sigma_j^2).$$

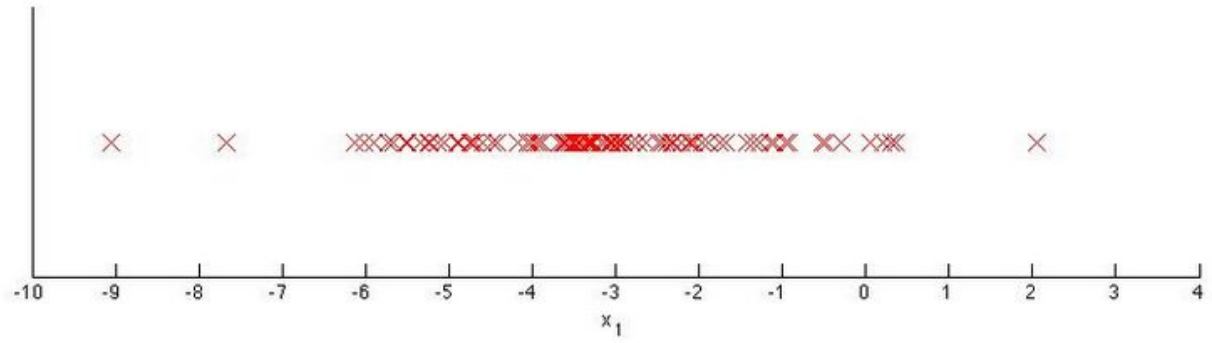
You have two features, x_1 = vibration intensity, and x_2 = heat generated. Both x_1 and x_2 take on values between 0 and 1 (and are strictly greater than 0), and for most "normal" engines you expect that $x_1 \approx x_2$. One of the suspected anomalies is that a flawed engine may vibrate very intensely even without generating much heat (large x_1, small x_2), even though the particular values of x_1 and x_2 may not fall outside their typical ranges of values. What additional feature x_3 should you create to capture these types of anomalies:

- $x_3 = x_1 + x_2$
- $x_3 = 1/x_2$
- $x_3 = x_1/x_2$
- $x_3 = 1/x_1$

Q.4. Which of the following are true? Check all that apply.

- In anomaly detection, we fit a model p(x) to a set of negative (y=0) examples, without using any positive examples we may have collected from previously observed anomalies.
- When evaluating an anomaly detection algorithm on the cross-validation set (containing some positive and some negative examples), classification accuracy is usually a good evaluation metric to use.
- In a typical anomaly detection setting, we have a large number of anomalous examples, and a relatively small number of normal/non-anomalous examples.
- When developing an anomaly detection system, it is often used to select an appropriate numerical performance metric to evaluate the effectiveness of the learning algorithm.

Q.5. You have a 1-D dataset $\{x^{(1)}, \ldots, x^{(m)}\}$ and you want to detect outliers in the dataset. You first plot the dataset and it looks like this:



Suppose you fit the gaussian distribution parameters μ_1 and σ_1^2 to this dataset. Which of the following values for μ_1 and σ_1^2 might you get?

- $\mu_1 = -3, \sigma_1^2 = 4$
- $\mu_1 = -6, \sigma_1^2 = 4$
- $\mu_1 = -3, \sigma_1^2 = 2$
- $\mu_1 = -6, \sigma_1^2 = 2$

Recommender Systems

Q.1. Suppose you run a bookstore, and have ratings (1 to 5 stars) of books. Your collaborative filtering algorithm has learned a parameter vector $\theta^{(j)}$ for user j, and a feature vector $x^{(i)}$ for each book. You would like to compute the "training error", meaning the average squared error of your system's predictions on all the ratings that you have gotten from your users. Which of these are correct ways of doing so (check all that apply)?

For this problem, let m be the total number of ratings you have gotten from your users. (Another way of saying this is that, $m = \sum_{i=1}^{n_m} \sum_{j=1}^{n_u} r(i,j)$).

[Hint: Two of the four options below are correct.]

$$\frac{1}{m} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left(\sum_{k=1}^{n} (\theta^{(j)})_k x_k^{(i)} - y^{(i,j)} \right)^2$$

$$\frac{1}{m} \sum_{(i,j):r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2$$

$$\frac{1}{m} \sum_{j=1}^{n_u} \sum_{i:r(i,j)=1} \left(\sum_{k=1}^{n} (\theta^{(k)})_j x_i^{(k)} - y^{(i,j)} \right)^2$$

$$\frac{1}{m} \sum_{(i,j):r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - r(i,j) \right)^2$$

Q.2. In which of the following situations will a collaborative filtering system be the most appropriate learning algorithm (compared to linear or logistic regression)?

- You manage an online bookstore and you have the book ratings for many users. For each user, you want to recommend other books she will enjoy, based on her own ratings and the ratings of other users.

- You manage an online bookstore and you have the book ratings from many users. You want to learn to predict the expected sales volume (number of books sold) as a function of the average rating of a book.

- You have written a piece of software that has downloaded news articles from many news websites. In your system, you also keep track of which articles you personally like vs. dislike, and the system also stores away features of these articles (e.g. word counts, name of author). Using this information, you want to build a system to try to find additional new articles that you personally will like.

- You run an online news aggregator, and for every user, you know some subset of articles that the user likes and some different subset that the user dislikes. You'd want to use this to find other articles that the user likes.

Q.3. You run a movie empire, and want to build a movie recommendation system based on collaborative filtering. There were three popular review websites (which we'll call A, B and C) which users go to rate movies, and you have just acquired all three companies that run these websites. You'd like to merge the three companies' datasets together to build a single/unified system. On website A, users rank a movie as having 1 through 5 stars. On website B, users rank on a scale of 1 - 10, and decimal values (e.g., 7.5) are allowed. On website C, the ratings are from 1 to 100. You also have enough information to identify users/movies on one website with users/movies on a different website. Which of the following statements is true?

- It is not possible to combine these websites' data. You must build three separate recommendation systems.
- You can combine all three training sets into one without any modification and expect high performance from a recommendation system.
- You can merge the three datasets into one, but you should first normalize each dataset's ratings (say rescale each dataset's ratings to a 1-100 range).
- Assume that there is at least one movie/user in one database that doesn't also appear in a second database, there is no sound way to merge the datasets, because of the missing data.

Q.4. Which of the following are true of collaborative filtering systems? Check all that apply.

- For collaborative filtering, the optimization algorithm you should use is gradient descent. In particular, you cannot use more advanced optimization algorithms. (L-BFGS/conjugate gradient/etc.) for collaborative filtering, since you have to solve for both the $x^{(i)}$'s and $\theta^{(j)}$'s simultaneously.
- For collaborative filtering, it is possible to use one of the advanced optimization algorithms (L-BFGS/conjugate gradient/etc.) to solve for both the $x^{(i)}$'s and $\theta^{(j)}$'s simultaneously.
- Suppose you are writing a recommender system to predict a user's book preferences. In order to build such a system, you need that user to rate all the other books in your training set.
- Even if each user has rated only a small fraction of all of your products (so r(i, j) = 0 for the vast majority of (i, j) pairs), you can still build a recommender system by using collaborative filtering.

Q.5. Suppose you have two matrices A and B, where A is 5x3 and B is 3x5. Their product is C=AB, a 5x5 matrix. Furthermore, you have a 5x5 matrix R where every entry is 0 or 1. You want to find the sum of all elements C(i, j) for which the corresponding R(i, j) is 1, and ignore all elements C(i,j) where R(i, j)=0. One way to do so is the following code:

```
C = A * B;
total = 0;
for i = 1:5
  for j = 1:5
    if (R(i,j) == 1)
      total = total + C(i,j);
    end
  end
end
```

Which of the following pieces of Octave code will also correctly compute this total? Check all that apply. Assume all options are in code.

- total = sum(sum((A * B) . * R))
- C = (A * B) . * R; total = sum(C(:));
- total = sum(sum(A * B) * R));
- C = (A * B) * R; total = sum(C(:));