MACS30000 Assignment 6

Dr. Richard Evans

Submitted by: Junho Choi

Due November 19, 2018 (11:30 AM)

Problem 1

(a) Before describing the criterion function, I will elaborate on how Netflix divided its dataset for the contest. The entire dataset was broken down into training set (which was further subdivided into probe and non-probe sets), test set, and quiz set. The training set was the one distributed to the contest participants for the purpose of constructing their models, and the test set was used for evaluating the model and decide who the winners are. The quiz set was used for calculating the leaderboard. (Netflix 2009). It seems that the division between test and quiz sets were made so that no participating team could deliberately tailor its model to a certain subset of the entire dataset.

Using such dataset, the challengers' objective was to create a model that could improve (i.e. reduce) the root mean squared error (abbreviated as RMSE) of prediction (Netflix 2009). The baseline algorithm for which the submissions were to be compared with was Cinematch, an algorithm then-used by Netflix. Cinematch yielded a RMSE of 0.9525, and Netflix was ambitious; it had set the upper bound of RMSE to be produced by the models of contestants to be 0.8572 ($= 0.9 \times 0.9525$), which meant at least a 10% improvement in RMSE. Anything above this would not be qualified for the Grand Prize (Netflix 2009). In addition, it is described by Bell et al. (2010) that the RMSEs for contestants were to be "round[ed]... to four decimal places" and the "tiebreaker was submission time of the best entries" (p. 29).

(b) In the beginning of Netflix Prize contest, the method that was most frequently used was the nearest neighbors algorithm. According to a documentation by SciKit-Learn (2018), nearest neighbors algorithm as a *classifier* tries to provide classification for an observation or a data point (that is unlabelled) by looking at other data points near it that have already been labelled. In the case of Netflix Prize contest, the point is not to classify, but rather to provide a predicted rating for a movie; therefore, the algorithm would be providing "a weighted average... of similar items by the same user" (Bell et al. 2010, p. 25).

Because the definition of proximity for data points can be rather arbitrary, it is described that "[s]imilarity is measured via Pearson correlation, cosine similarity, or other metric calculated on the ratings" (Bell et al. 2010, p. 25). Yet it can still be critiqued that the choice of a certain metric over another is arbitrary (Bell et al. 2010, p. 26). In addition, Bell et al. (2010) further elaborate that problems may arise when there are two highly-correlated neighbors of a data point or when a data point has only a small number of close neighbors (p. 26).

(c) According to Bell et al. (2010), the one of the most successful models to use by itself or in coordination with others was matrix factorization, which is one of the "latent factor models" (p. 26). A latent factor model, as hinted by its name, is one that is able to derive "hidden" structures behind a dataset "without requiring external information" (Bell et al. 2010, p. 27). Matrix factorization, among such latent factor models, seems to be particularly malleable in that there are multiple ways to optimize for the objective function; one of such is stochastic gradient descent, which does not use (relatively) significantly high computational power due to "avoiding the repeated inversion of large matrices" (Bell et al. 2010, p. 27). Such characteristics – described as "power and flexibility" by Bell et al. (2010) – seem to be the reason why matrix factorization was able to be effectively blended with other models to improve predictions (p. 27).

Problem 2

(a) I have successfully created my account for Project Euler, and my username and friend key are as follows:

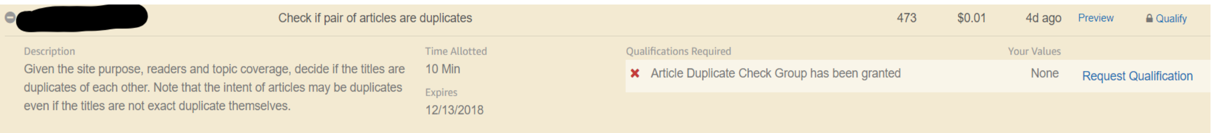
kildar0link: `1407406_YdiYrhyIOSrS1UbAAgHE3ykCgKl3dAoR`

(b) For this sub-question, I have solved Problem 2 in Project Euler, called "Even Fibonacci numbers." I have successfully solved this question using Python and the code for this is given in the file `junhoc_A6.ipynb` (and its `.pdf` counterpart `junhoc_A6_ipynb.pdf`).

(c) There are many achievements on the Project Euler website, and many of them have to do with solving problems. I have, at the moment, received the `Baby Steps` achievement which have to do with solving three problems. In the future, I would like to receive the following achievements: `Fibonacci Fever`, `Triangle Trophy`, and `Prime Obsession`. These achievements are interesting in that they have to do with not only solving a certain amount of problems, but also solving problems with specifically-designated numbers. For instance, `Fibonacci Fever` has to do with solving problems with IDs that are Fibonacci-numbered (hence the name).

Problem 3

Figure 1: Amazon Mechanical Turk Human Computation Project Details



[redacted] Check if pair of articles are duplicates 473 $0.01 4d ago

Description
Given the site purpose, readers and topic coverage, decide if the titles are duplicates of each other. Note that the intent of articles may be duplicates even if the titles are not exact duplicate themselves.

Time Allotted
10 Min

Expires
12/13/2018

Qualifications Required	Your Values	
✗ Article Duplicate Check Group has been granted	None	Request Qualification

(a) For this assignment, I have chosen a HIT (that is a human computation project) that asks participants to check whether two articles are duplicates of one another. Figure 1 shows some of the (simpler) details to the aforementioned HIT. Note that the requester's name has been, once again, crossed out in case there arises any problems.

(b) Unlike the experiment HIT that I was able to find for the last problem set, there are no further details to the payment structure other than the up-front reward of \$0.01. While I do suspect that there can be additional rewards to this HIT, I will have to conclude here as I do not have any further information.

(c) The only qualification that is required is the one that is called "Article Duplicate Check Group." There are no tests to complete for this qualification, but it seems that it can be met after sending a request for approval to the HIT requester.

(d) The allotted time for this task is 10 minutes, according to the description. Because there are not any further description to the task, I am not able to answer about the number of items.[1] The implied hourly rate is \$0.06 per hour (= 0.01 (dollars)/(1/6) (hours)), given that the allotted time is approximately the same as the time it takes to complete the HIT.

(e) The said HIT is said to expire on **December 13, 2018**, which is around 26 days from November 17, 2018 (which is the day I started to work on this assignment).

(f) Given that there are not any further structure to the task, the requester would paying \$0.01 per participant. The Amazon Mechanical Turk website notes that there is a "20% fee on the reward and bonus amount (if any) you pay Workers" (Amazon Mechanical Turk n.d.). Therefore, the requester would actually be facing the cost of \$0.012 per participant. Given that there are 1,000,000 participants, the total cost would be \$12,000.

Problem 4

(a) I have successfully created a Kaggle account and have logged into the website.

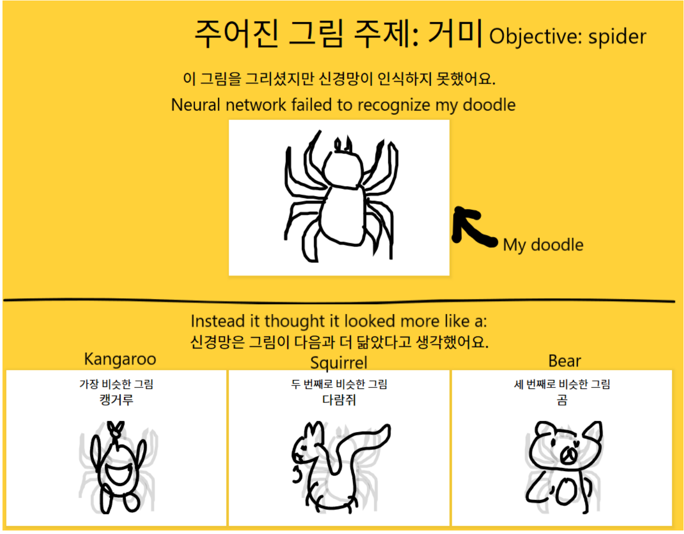
(b) For this question, I have chosen to cover a Kaggle challenge called **Quick, Draw! Doodle Recognition Challenge**, sponsored by **Google AI**. Google AI is, as its namesake, an artificial intelligence research team at Google. According to its website, the team seeks to provide new perspectives to both old and upcoming problems with the intent to make everyone's lives more efficient (Google AI n.d.).

Before moving further, I will describe the proposed objective of this challenge. The given task of this challenge is to build a classifier for the "Quick, Draw!"

[1] Note that I was able to preview the task, but I was able to see only one task item. This further complicates my ability to answer the question about the number of items I could complete.

dataset that performs "better."[2] Quick, Draw! is a game developed by Google that showcases one way how neural networks work. For each trial of the game, a participant is given an objective to draw and 20 seconds to complete the task. Neural networks may or may not be able to classify a certain doodle drawn by a game participant; for instance, I was tasked to draw a spider, but instead neural networks classified my drawing to be more akin to a kangaroo, a squirrel, or a bear (see Figure 2 for this example).

Figure 2: Google Quick, Draw! Example



주어진 그림 주제: 거미 Objective: spider

이 그림을 그리셨지만 신경망이 인식하지 못했어요.

Neural network failed to recognize my doodle

Instead it thought it looked more like a:

신경망은 그림이 다음과 더 닮았다고 생각했어요.

Note: I played the Korean version of the game,
so I provided the English translations to help understanding.

The Challenge describes that submissions will be evaluated based on how well a submission classifies the doodles, and to avoid any ambiguity it uses mean average precision at 3 (MAP@3) as its metric. The equation for MAP@3 is given as:

$$MAP@3 = \frac{1}{U}\sum_{u=1}^{U}\sum_{k=1}^{\min(n,3)} P(k)$$

in which "U is the number of scored drawings in the test data, $P(k)$ is the precision cutoff at k, and n is the number of predictions per drawing" (Kaggle 2018). To interpret the equation, it seems that any program has up to three "guesses" to figure out what the drawing is about. There are U doodles in the data with scores attached, and therefore the program is required not just to perform well for correctly classifying one picture but many pictures in general.

The team with a submission yielding the highest MAP@3 score will be the winner, and there are prizes not only for the first place ($12,000) but also second and third places ($8,000 and $5,000, respectively) as well (Kaggle 2018).

[2] The metric for evaluating how good a classifier is will be elaborated shortly.

While one may think that there are no issues with the honor code for this challenge, there may be potential problems with sharing or selling codes. For instance, a high-ranker on the leaderboard might be willing to sell one's code for a monetary return. Or, a group of high-rankers might be willing to collude by sharing codes so that they can safely dominate the leaderboard and share the prize for themselves. To prevent such disasters, there are guidelines under the "Rules" section that tell participants about things such as not privately sharing codes, number of members per team, and so forth (Kaggle 2018). One thing that does make me worry, however, is the fact that there seems to be no way to enforce these rules.

Timeline for this challenge is as follows. November 27, 2018 is the last day on which any potential participant can commit to participating, after accepting the aforementioned guidelines in the "Rules" section. November 28, 2018 is the last day on which participating teams can merge different teams, or individual participants can choose to find a team to join. Finally, December 4, 2018 is the last day that the participants can submit (Kaggle 2018).

Submissions can be made from Kaggle Kernels, which is hosted by Kaggle itself. Any team member (once granted permission) can access the kernel to revise or view the code without exposing it to outsiders (Kaggle 2018).

(c) In the case of Quick, Draw! Doodle Recognition Challenge, the sponsor (being Google AI) is very straightforward with its proposed use of the winning submission data (and potentially, other non-winner submission data as well). Google AI mentions in its description to the challenge that it would be able to improve research in handwriting recognition and potentially extend its use to areas such as "OCR (Optical Character Recognition), ASR (Automatic Speech Recognition), [and] NLP (Natural Language Processing)" (Kaggle 2018). Ultimately, I think that these data will be used for various products to be developed or in development by Google, as Google AI is said to help further improve Google products and programs such as Google Assistant, Google Translate, and more (Google AI n.d.).

References

Amazon Mechanical Turk. n.d. "Amazon Mechnical Turk Pricing." https://requester.mturk.com/pricing.

Bell, Robert M., Yehuda Koren, and Chris Volinsky. 2010. "All Together Now: A Perspective on the Netflix Prize." *Chance* 23 (1): 24-29.

Google AI. n.d. "About: Bringing the Benefits of AI to Everyone." https://ai.google/about/.

Kaggle. 2008. "Quick, Draw! Doodle Recognition Challenge: How Accurately Can You Identify a Doodle?"

https://www.kaggle.com/c/quickdraw-doodle-recognition#evaluation.

Netflix. 2009. "The Netflix Prize Rules." https://www.netflixprize.com/rules.html.

SciKit-Learn. 2018. "Nearest Neighbors." https://scikit-learn.org/stable/modules/neighbors.html.