**DAY-1 [17-09-2021]: Introduction**

1. Matrix completion: Sometimes we have missing data, that is, variables whose values are unknown. For example, we might have conducted a survey, and some people might not have answered certain questions. Or we might have various sensors, some of which fail. The corresponding design matrix will then have “holes” in it; these missing entries are often represented by NaN, which stands for “not a number”. The goal of imputation is to infer plausible values for the missing entries. This is sometimes called matrix completion. Below we give some example applications
2. Image Impainting: An interesting example of an imputation-like task is known as image inpainting. The goal is to “fill in” holes (e.g., due to scratches or occlusions) in an image with realistic texture. This is illustrated in Figure 1.12, where we denoise the image, as well as impute the pixels hidden behind the occlusion. This can be tackled by building a joint probability model of the pixels, given a set of clean images, and then inferring the unknown variables (pixels) given the known variables (pixels). This is somewhat like masket basket analysis, except the data is real-valued and spatially structured, so the kinds of probability models we use are quite diferent. See Sections 19.6.2.7 and 13.8.4 for some possible choices.
3. Collaborative filtering: Another interesting example of an imputation-like task is known as collaborative filtering. A common example of this concerns predicting which movies people will want to watch based on how they, and other people, have rated movies which they have already seen. The key idea is that the prediction is not based on features of the movie or user (although it could be), but merely on a ratings matrix. More precisely, we have a matrix X where X(m, u) is the rating (say an integer between 1 and 5, where 1 is dislike and 5 is like) by user u of movie m. Note that most of the entries in X will be missing or unknown, since most users will not have rated most movies. Hence we only observe a tiny subset of the X matrix, and we want to predict a different subset. In particular, for any given user u, we might want to predict which of the unrated movies he/she is most likely to want to watch. In order to encourage research in this area, the DVD rental company Netflix created a competition, launched in 2006, with a $1M USD prize (see http://netflixprize.com/). In particular, they provided a large matrix of ratings, on a scale of 1 to 5, for ∼ 18k movies created by ∼ 500k users. The full matrix would have ∼ 9 × 109 entries, but only about 1% of the entries are observed, so the matrix is extremely sparse. A subset of these are used for training, and the rest for testing, as shown in Figure 1.13. The goal of the competition was to predict more accurately than Netflix’s existing system. On 21 September 2009, the prize was awarded to a team of researchers known as “BellKor’s Pragmatic Chaos.
4. Market Basket Analysis: In commercial data mining, there is much interest in a task called market basket analysis. The data consists of a (typically very large but sparse) binary matrix, where each column represents an item or product, and each row represents a transaction. We set xij = 1 if item j was purchased on the i’th transaction. Many items are purchased together (e.g., bread and butter), so there will be correlations amongst the bits. Given a new partially observed bit vector, representing a subset of items that the consumer has bought, the goal is to predict which other bits are likely to turn on, representing other items the consumer might be likely to buy. (Unlike collaborative filtering, we often assume there is no missing data in the training data, since we know the past shopping behavior of each customer.)
5. No Free Lunch Theorem: The No Free Lunch Theorem, often abbreviated as NFL or NFLT, is a theoretical finding that suggests all optimization algorithms perform equally well when their performance is averaged over all possible objective functions.

Resources:

1. https://machinelearningmastery.com/distance-measures-for-machine-learning/