Quiz: Notes

Question 1:

Which of the following problems would you approach with an anomaly detection algorithm (rather than a supervised learning algorithm)? Check all that apply.

- [x] You run a power utility (supplying electricity to customers) and want to monitor your electric plants to see if any one of them might be behaving strangely.

Correcto

- [ ] You run a power utility and want to predict tomorrow's expected demand for electricity (so that you can plan to ramp up an appropriate amount of generation capacity).

Deseleccionado es lo correcto

- [x] A computer vision / security application, where you examine video images to see if anyone in your company's parking lot is acting in an unusual way.

Correcto

- [ ] A computer vision application, where you examine an image of a person entering your retail store to determine if the person is male or female.

Deseleccionado es lo correcto

Question 2:

In our notation, $r(i,j) = 1$ if user j has rated movie i, and $y^{(i,j)}$ is his rating on that movie. Consider the following example (no. of movies $n_m = 2$, no. of users $n_u = 3$):

.	User 1	User 2	User 3
Movie 1	0	1	?
Movie 2	?	5	5

What is $r(2,1)$? How about $y^{(2,1)}$?

- ( ) $r(2,1) = 0,\ y^{(2,1)} = 1$
- ( ) $r(2,1) = 1,\ y^{(2,1)} = 1$
- (•) $r(2,1) = 0,\ y^{(2,1)} = \text{undefined}$

Correcto

- ( ) $r(2,1) = 1,\ y^{(2,1)} = \text{undefined}$

Question 3:

Consider the following movie ratings:

.	User 1	User 2	User 3	(romance)
Movie 1	0	1.5	2.5	?

Note that there is only one feature x_1. Suppose that:

$$\theta^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \theta^{(2)} = \begin{bmatrix} 0 \\ 3 \end{bmatrix}, \theta^{(3)} = \begin{bmatrix} 0 \\ 5 \end{bmatrix}$$

What would be a reasonable value for $x_1^{(1)}$ (the value denoted "?" in the table above)?

- (selected) 0.5

  Correcto

- 1
- 2
- Any of these values would be equally reasonable.

Question 4:

Suppose you use gradient descent to minimize:

$$\min_{x^{(1)},\ldots,x^{(n_m)}} \frac{1}{2} \sum_{i=1}^{n_m} \sum_{j:r(i,j)=1} \left((\theta^{(j)})^T x^{(i)} - y^{(i,j)} \right)^2 + \frac{\lambda}{2} \sum_{i=1}^{n_m} \sum_{k=1}^{n} (x_k^{(i)})^2$$

Which of the following is a correct gradient descent update rule for $i \neq 0$?

- $x_k^{(i)} := x_k^{(i)} + \alpha \left(\sum_{j:r(i,j)=1} \left((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)} \right) \theta_k^{(j)} \right)$
- $x_k^{(i)} := x_k^{(i)} - \alpha \left(\sum_{j:r(i,j)=1} \left((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)} \right) \theta_k^{(j)} \right)$
- $x_k^{(i)} := x_k^{(i)} + \alpha \left(\sum_{j:r(i,j)=1} \left((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)} \right) \theta_k^{(j)} + \lambda x_k^{(i)} \right)$
- (selected) $x_k^{(i)} := x_k^{(i)} - \alpha \left(\sum_{j:r(i,j)=1} \left((\theta^{(j)})^T (x^{(i)}) - y^{(i,j)} \right) \theta_k^{(j)} + \lambda x_k^{(i)} \right)$

  Correcto

Question 5:

In the algorithm we described, we initialized $x^{(1)}, \ldots, x^{(n_m)}$ and $\theta^{(1)}, \ldots, \theta^{(n_u)}$ to small random values. Why is this?

- ○ This step is optional. Initializing to all 0's would work just as well.
- ○ Random initialization is always necessary when using gradient descent on any problem.
- ○ This ensures that $x^{(i)} \neq \theta^{(j)}$ for any i, j.
- ◉ This serves as symmetry breaking (similar to the random initialization of a neural network's parameters) and ensures the algorithm learns features $x^{(1)}, \ldots, x^{(n_m)}$ that are different from each other.

Correcto

Question 6:

Let $X = \begin{bmatrix} - & (x^{(1)})^T & - \\ & \vdots & \\ - & (x^{(n_m)}) & - \end{bmatrix}$, $\Theta = \begin{bmatrix} - & (\theta^{(1)})^T & - \\ & \vdots & \\ - & (\theta^{(n_u)}) & - \end{bmatrix}$.

What is another way of writing the following:

$$\begin{bmatrix} (x^{(1)})^T(\theta^{(1)}) & \ldots & (x^{(1)})^T(\theta^{(n_u)}) \\ \vdots & \ddots & \vdots \\ (x^{(n_m)})^T(\theta^{(1)}) & \ldots & (x^{(n_m)})^T(\theta^{(n_u)}) \end{bmatrix}$$

- ○ $X\Theta$
- ○ $X^T\Theta$
- ◉ $X\Theta^T$

Correcto

- ○ $\Theta^T X^T$

Question 7:

We talked about mean normalization. However, unlike some other applications of feature scaling, we did not scale the movie ratings by dividing by the range (max – min value). This is because:

- [ ] This sort of scaling is not useful when the value being predicted is real-valued.
- [x] All the movie ratings are already comparable (e.g., 0 to 5 stars), so they are already on similar scales.

  Correcto

- [ ] Subtracting the mean is mathematically equivalent to dividing by the range.
- [ ] This makes the overall algorithm significantly more computationally efficient.