INF554 - Machine Learning I

Lab 3: Answers for Questions

Coding Tasks

See `Lab3_solutions.ipynb`.

Question 1

From the *lab3* notes, we saw that the *likelihood* is:

$$L(\theta) = \prod_{i=1}^{n} \sigma(\theta^\top \mathbf{x}_i)^{y_i} (1 - \sigma(\theta^\top \mathbf{x}_i))^{(1-y_i)} \tag{1}$$

We obtain the log-likelihood by simply applying the log function:

$$LL(\theta) = \sum_{i=1}^{n} y_i \log \sigma(\theta^\top \mathbf{x}_i) + (1 - y_i) \log \left[1 - \sigma(\theta^\top \mathbf{x}_i)\right] \tag{2}$$

The cost function is obtained as $E(\theta) = -LL(\theta)$:

$$E(\theta) = -\sum_{i=1}^{n} y_i \log \sigma(\theta^\top \mathbf{x}_i) + (1 - y_i) \log \left[1 - \sigma(\theta^\top \mathbf{x}_i)\right] \tag{3}$$

In order to calculate the gradient of this function, let's denotate $z_i = \theta^\top \mathbf{x}_i$ and $p_i = \sigma(\theta^\top \mathbf{x}_i)$; then, thanks to the *chain rule*, we can calculate the partial derivatives as follows:

$$\frac{\partial E(\theta_\mathbf{j})}{\partial \theta_\mathbf{j}} = \frac{\partial E(\theta_\mathbf{j})}{\partial p} \cdot \frac{\partial p}{\partial \theta_\mathbf{j}} = \frac{\partial E(\theta_\mathbf{j})}{\partial p} \cdot \frac{\partial p}{\partial z} \cdot \frac{\partial z}{\partial \theta_\mathbf{j}}$$

where $\frac{\partial p}{\partial z} = \sigma(z)(1 - \sigma(z))$ (derivative of sigmoid) and $\frac{\partial z}{\partial \theta_\mathbf{j}} = x_j$ as only x_j interacts with θ_j.

We can now calculate the gradient of the cost function:

$$
\begin{aligned}
\nabla_\theta E(\theta) &= -\nabla_\theta \left[\sum_{i=1}^{n} y_i \log p_i + (1 - y_i) \log(1 - p_i) \right] \\
&= -\sum_{i=1}^{n} \left[\frac{y_i}{p_i} \nabla_\theta p_i + \frac{(1 - y_i)}{(1 - p_i)} \nabla_\theta (1 - p_i) \right] \\
&= -\sum_{i=1}^{n} \left[y_i (1 - p_i) \nabla_\theta z_i - (1 - y_i) p_i \nabla_\theta z_i \right] \\
&= -\sum_{i=1}^{n} \left[y_i (1 - p_i) x_i - (1 - y_i) p_i x_i \right] \\
&= -\sum_{i=1}^{n} \left[y_i (1 - p_i) - (1 - y_i) p_i \right] x_i \\
&= -\sum_{i=1}^{n} \left[y_i - p_i y_i - p_i + p_i y_i \right] x_i \\
&= -\sum_{i=1}^{n} \left[y_i - p_i \right] x_i \\
&= -\sum_{i=1}^{n} \left[y_i - \sigma(\theta^\top \mathbf{x}_i) \right] x_i
\end{aligned}
\tag{4}
$$

Question 2

From the graphs we can see that the 0.5 learning rate is too large and it looks like we're overstepping the minimum. The result obtained with learning rates 0.05 and 0.005 are similar but it can be seen that convergence is slower for the smaller learning rate.

Since the error function is convex, we can converge to a global minimum with a non-zero learning rate. However, the gradient is not guaranteed to always pointing towards the minimum (as it can be observed by the large learning rate examples above).

Question 3

The decision boundary learned with the greatest learning rate is suboptimal. The boundaries learned with the other learning rates are almost undistinguishable. This confirms what we saw during training and in the previous graphs.

Question 4

SGD is less costly than GD, with classical GD you have to compute the gradient over all training data if you want to have a consistent estimator of the true gradient, whereas with SGD you have a consistent estimator with few computations. Moreover SGD introduces noise via the variance of the estimator, shielding the model from overfitting. Finally, SGD can, and will always eventually, escape from local minima.

Question 5

Small k tends to overfit. However, if k is too high, we run the risk of under-fitting.

Question 6

$O(ndk)$: obviously an issue for large datasets with many instances n or features d.