CSCI-UA 480 Robot Intelligence Homework 2

Chuanyang Jin

February 21, 2023

Question 1: Optimization Roulette (16 points per part)

For each of the stated problems below, identify which of the following is the most appropriate method to employ: (1) exact solve; (2) linear/logistic regression; (3) neural network trained with gradient descent; or (4) cross-entropy method. Describe how precisely you would employ this method to solve the problem, and explain your reasoning for why your chosen method is the most appropriate. In some settings, two methods may be equally appropriate; if you think this is the case, identify the pair, explain why both methods are equally effective, and elucidate how both can be used to solve the problem. You will be graded based on the quality and correctness of your answers and accompanying explanations.

Part (a): Boolean Satisfiability. Let $x_1, ..., x_n$ be a set of boolean variables, i.e. $x_i \in \{\text{TRUE}, \text{FALSE}\}$, $\forall i : 1 \leq i \leq n$. Any boolean formula ϕ over variables $x_1, ..., x_n$ can be written as set of boolean or-clauses joined together with the "AND" operation; for example, $(x_1 \vee x_2 \vee x_3) \wedge (x_1 \vee x_n)$. We say that a boolean formula is satisfiable if there exists an assignment to its variables such that all or-clauses evaluate to TRUE. Given a fixed boolean formula $\phi(x_1, ..., x_n)$, your goal is to use an optimization method to efficiently and exactly determine whether it is is satisfiable.

Solution.

Choice: exact solve.

Exact solve method can guarantee finding the correct answer. For a fixed boolean formula, we can examine every possibility one by one. Given a possible assignment to the variables, we can evaluate it with the rules of "AND", "OR", and "Negation". If all possibilities do not evaluate to TRUE, the formula is not satisfiable; otherwise, the formula is satisfiable.

Linear/logistic regression, neural network, or cross-entropy method is not appropriate in this problem, since they cannot guarantee finding the correct answer. They are more appropriate for problems such as prediction, classification and optimization. □

Part (b): Data-Limited Binary Classification. You are a newly hired engineer working for the self-driving startup Druise. During your hiring interview, you told your manager that you took "Robot Intelligence" in college, and so they ask you to put your skills to the test by developing a "STOP" sign classifier, i.e. a function which given an image $x \in X$, predicts whether or not the image contains a "STOP" sign. Unfortunately, the startup has limited resources to collect data, so you are given a dataset with only five hundred images $x_1, ..., x_{500}$, each image x_i accompanied by the label $y_i : y_i \in \{\text{"STOP SIGN"}, \text{"NOT A STOP SIGN"}\}$.

Solution.

Choice: linear/logistic regression, or the neural network (with techniques such as data augmentation, regularization, and early stop).

Linear/logistic regression may solve the data-limited binary classification problem. We can turn the color of pixels or small patches into features and perform the regression. However, there is a risk that the regression may not be sufficient to capture the complex relationships between the image features and the sign labels.

The neural network may also be approximate. But with only 500 labeled images, there may be a risk of overfitting, where the model learns to fit the training data too closely and fails to generalize well to new, unseen data. Therefore, techniques such as data augmentation, regularization, and early stopping may need to be used to mitigate overfitting.

The exact solve does not exist for this problem.

The cross-entropy method more suitable for optimization problems where the objective function is not easily differentiable or not known in closed form, and the solution space is large, complex, and discrete. It is hard to use the method to find the parameters in this problem. □

Part (c): Big Data Multi-Class Classification. You were successful at producing a decent "STOP" sign classifier at Druise, and have since left the company to join the force of its bitter rival, Vaymo. Your new manager asks you to produce a general-purpose road sign classifier using a massive dataset containing one hundred-million road sign images, $x_1, ..., x_{1e8}$, each image x_i accompanied by a label $y_i : y_i \in \{$"STOP SIGN", "YIELD STOP SIGN", "SLIPPY ROAD SIGN"$\}$.

Solution.

Choice: the neural network.

We can use a deep convolutional neural network (CNN) architecture, which is well-suited for image classification tasks. The CNN would consist of several convolutional layers, followed by pooling layers and fully connected layers. The output layer would have three nodes, corresponding to the three classes of road signs.

Linear/logistic regression can only model linear decision boundaries between classes in the feature space, which may not be sufficient to capture the complex relationships between the image features and the sign labels.

The exact solve does not exist for this problem.

The cross-entropy method more suitable for optimization problems where the objective function is not easily differentiable or not known in closed form, and the solution space is large, complex, and discrete. It is hard to use the method to find the parameters in this problem. □

Part (d): Watching a Rocket Launch. The famed rocket company GreenOrigin is launching their brand new rocket from a launchpad in the suburbs of New York City. You are planning to watch the launch. Ideally, you would like to see the rocket in air for as long as possible, and you are willing to go at most 0.1 degree north or south in latitude, as well as at most 0.1 degree east or west in longitude from the geographical coordinates of your dorm to do so. Lucky for you, GreenOrigin has released their visibility model for the launch; indeed, given a pair (x, y) of latitude and longitude coordinates in New York City, the rocket is estimated to be visible for $t(x, y) = |x - 40.7| \cdot |(x - 40.7) + (y + 73.9)^4 + \sin^2(x/40.7)|$ minutes. You would like to determine the exact coordinates satisfying your constraints at which the rocket is visible for the maximal amount of time.

Solution.

Choice: the cross-entropy method.

We can use the cross-entropy method. We can iteratively samples input coordinates from a distribution and updates the distribution based on the performance of the samples.

The function is non-linear, and the constraints are also non-linear, which makes it difficult to find the optimal solution using an exact solve.

Linear/logistic regression is not suitable for this problem since the function is non-linear, and the regression model can only capture linear relationships between the input and output variables.

The neural network is not suitable for this problem as it is a mathematical optimization problem. It is hard to formulate the loss objective for training. □

Part (e): Plant Watering. A friend gifts you a mysterious house plant. You're not sure how often and how much it needs to be watered. You decide to launch a long-term experiment using an optimization mechanism discussed in class to find a near-optimal weekly watering strategy, $\pi : \{1, ..., 7\} \to \{0, \frac{1}{2}, 1\}$, mapping each day of the week to one of three quantities of water: no cups, half a cup, or a full cup. At the end of each week, you will take a photo of the plant, record how healthy it looks on the scale of 1 to 10, and possibly adjust your strategy for the following week.

Solution.

Choice: the cross-entropy method.

The cross-entropy method is a suitable optimization method for finding near-optimal solutions in situations where there is uncertainty about the effectiveness of different strategies, which is the case in this problem. The cross-entropy method would allow for the exploration of different watering strategies and the evaluation of their effectiveness in growing a healthy plant.

Since we don't have the data of strategy-result pairs, it is hard to find the exact solve, use linear/logistic regression, or train neural networks. □

Question 2: GD and SGD (10 points per part)

Respond to each of the stated questions in parts (a) and (b). You may keep your answers concise.

Part (a): Learning Rate. What is the role of learning rate in gradient descent? What happens if the learning rate is set too low? What happens if it is set too high?

Solution.

In gradient descent, the learning rate is a hyperparameter that determines the step size at each iteration while updating the parameters of the model.

If the learning rate is set too low, the model will take too small steps at each iteration, which can lead to slow convergence. The optimization process may take too long to converge to a minimum, and the model may get stuck in a suboptimal local minimum.

If the learning rate is set too high, the model will take too large steps at each iteration. It can cause the optimization process to overshoot the minimum and diverge, or bounce back and forth across the minimum.

□

Part (b): Batch Size. What is the role of batch size in stochastic gradient descent? In what situation might we need to lower the batch size? Increase it?

Solution.

In stochastic gradient descent, the batch size is the number of samples used to compute the loss and update the parameters in each iteration.

In situations where the data is noisy or there is a lot of variance in the data, a smaller batch size may be needed to avoid getting stuck in local minima. When the model is overfitting, lowering the batch size can also introduce more noise and regularization to the optimization process. Additionally, smaller batch sizes can be useful when the available memory is limited.

Conversely, in situations where the data is relatively clean or there is less variance, a larger batch size improve the computational efficiency. It is also helpful where the model is underfitting and requires more data to generalize well.

□