CS589: Machine Learning - Fall 2017

Scenario Template and Examples

Generic scenario template

Take a **[classification / regression]** dataset $(x_1, y_1), ..., (x_N, y_N)$ where $x \in \mathbb{R}^p$ and a corresponding validation set $(x_1, y_1), ..., (x_M, y_M)$. **[further details about dataset]**. You train a **[K-NN / tree / linear / neural network]** model using **[MSE / MAE / Hinge / 0-1 / logistic]** loss with **[ridge / lasso / no]** regularization and **[further details about training procedure]**. After training, you observe a mean training **[MSE / MAE / Hinge / 0-1 / logistic]** loss of **[some number]** and a mean validation **[MSE / MAE / Hinge / 0-1 / logistic]** loss of **[some number]**. (Regularization penalties are not included in these losses.)

Now, you **[change the training data somehow / change the loss somehow / change the training procedure somehow / change nothing]**.

Is it possible to now observe a mean **[train / validation]** **[MSE / MAE / Hinge / 0-1 / logistic]** loss of **[some number]**? **Possible** / **Not possible**. (Circle one)

1. First Example

Take a classification dataset $(x_1, y_1), ..., (x_N, y_N)$ where $x \in \mathbb{R}^p$ and a corresponding validation set $(x_1, y_1), ..., (x_M, y_M)$. The output is binary, with $y_i \in \{-1, +1\}$, and you have $N = 100$ and $p = 5$. You train a linear model using hinge loss with ridge regularization and a ridge penalty of $\lambda = 0.1$. After training, you observe a mean training hinge loss of 0.05 and a mean validation hinge loss of 0.09. (Regularization penalties are not included in these losses.)

Now, you re-train the model using a ridge penalty of $\lambda = 0.01$.

Is it possible to now observe a mean train hinge loss of 0.07? **Possible** / **Not possible**.

Is it be possible to now observe a mean validation loss of 0.07? **Possible** / **Not possible**.

2. Second example

Take a regression dataset $(x_1, y_1), ..., (x_N, y_N)$ where $x \in \mathbb{R}^p$ and a corresponding validation set $(x_1, y_1), ..., (x_M, y_M)$. You have $N = 100$ and $p = 1$, so each input is scalar with $x_i \in \mathbb{R}$. You train a regression tree model with a maximum depth of 3 using MSE loss. After training, you observe a mean training MSE loss of 1.25 and a mean validation MSE loss of 1.25.

Now, you perform a polynomial basis expansion of order 2, including a basis term. Thus, you create a new training set $(x'_1, y'_1), ...(x'_N, y'_N)$ with $x'_i = (1, x_i, x_i^2)$. The validation set is transformed in the same way. You re-train the model in exactly the same way, using a regression tree model with a maximum depth of 3 using MSE loss.

Is it possible to now observe a mean train MSE loss of 1.5? **Possible** / **Not possible**.

Is it possible to now observe a mean train MSE loss of 1.0? **Possible** / **Not possible**.

Is it possible to now observe a mean validation MSE loss of 1.5? **Possible** / **Not possible**.

Is it possible to now observe a mean validation MSE loss of 1.0? **Possible** / **Not possible**.

3. Third example

Take a classification dataset $(x_1, y_1), ..., (x_N, y_N)$ where $x \in \mathbb{R}^p$ and a corresponding validation set $(x_1, y_1), ..., (x_M, y_M)$. You have $N = 100$ and $p = 10$ and there are 3 classes. Thus, $y_i \in \{1, 2, 3\}$. You train a classification tree model with a maximum depth of 3 using the Gini entropy to determine splits. After training, you observe a mean training 0-1 loss of 0.15 and a mean validation 0-1 loss of 0.2.

Now, your client comes to you and says "oh, wait– I forgot one of the columns in the training data!". You are given an additional feature z_i for each input x_i. Thus, you create a new training set $(x'_1, y'_1), ...(x'_N, y'_N)$ with $x'_i = (x_i, z_i)$. The validation set is transformed in the same way. You re-train the model in exactly the same way (now using inputs $x'_i \in \mathbb{R}^{p+1}$).

Is it possible to now observe a mean train 0-1 loss of 0.10? **Possible / Not possible**.

Is it possible to now observe a mean train 0-1 loss 0.20? **Possible / Not possible**.

Hints for these problems

- What's the point of these problems? In the real world, machine learning systems often exhibit surprising behavior. It is often difficult to diagnose the cause of this behavior, and huge amounts of time can be spent looking for the cause in the wrong place. Understanding what is possible in scenarios like these is enormously helpful to being able to debug real-world systems.
- You may be given irrelevant information. Just because the problem tells you a specific value of N or p does not mean this information is relevant to the correct answer.
- Ways that things can change between the initial stage and the second stage: The data might be subjected to a basis expansion. The regularization constant might change. The regularizer might change (e.g. from ridge to lasso). A different training procedure might be used. Some detail of the procedure might change (e.g. Gini vs. entropy for classification trees). The evaluation in the second stage might use a different loss function.