Supervised Learning: a Business of Cost-Cutting

STUFF TO KNOW BY HEART - EVEN WHEN DRUNK!

*1. Supervised learning is about achieving **low Hypothesis-vs.-Target costs***

*2. **Cost functions** measure degree of prediction errors: low when errors are small, high when they are large*

*3. **Training** is mathematical optimization that makes cost decrease, i.e. make errors smaller*

In a supervised learning framework, we give the machine Input $\mathbf{X}$ - a matrix/array representing m cases of input features - and Target Output $\mathbf{Y}$ representing the m corresponding "**right answers**", and want the machine to learn a structured formula that transforms $\mathbf{X}$ to Hypothesized Output $\mathbf{H}$ that is similar/close to $\mathbf{Y}$.

The extent of $\mathbf{H}$'s similarity/closeness to $\mathbf{Y}$ - which is the criterion to judge how well a Supervised Learning Model learns to mimick $\mathbf{Y}$ from knowing $\mathbf{X}$ - is numerically measured by a certain specified scalar cost function $c(\mathbf{H}, \mathbf{Y})$. This function c's value should be small when $\mathbf{H}$ is very "similar" or "close" to $\mathbf{Y}$, and large otherwise. In almost all Supervised Learning Models in practical use nowadays, $c(\mathbf{H}, \mathbf{Y})$ is a continous real-valued function and its partial derivative $\frac{\partial c}{\partial \mathbf{H}}$ with respect to $\mathbf{H}$ is computable as a certain function $d(\mathbf{H}, \mathbf{Y})$. Cost functions are usually measured on an average per-case basis.

The task of helping a Supervised Learning Model learn - or so-called "**training**" it - involves mathematical optimization procedures that make the average per-case $\mathbf{H}$-vs.-$\mathbf{Y}$ cost decrease when we let the Model see more and more cases of inputs and corresponding "right-answer" target outputs. Once trained until its cost has decreased to an acceptably low level, a Model will make only small errors and hence be a good tool for predicting the output $\mathbf{y}$ from a not-yet-seen input $\mathbf{x}$.

NOTE ON OVER-FITTING

Note that we are using the rather gentle phrase "acceptably low" instead of the stronger word "minimized". This is because when a Supervised Learning Model does achieve the absolutely smallest possible error rate during the "training" process, it will have over-learned: not only will it have learned the overall rules of the game (which are useful when generalizing to new cases), it will have also **memorized various irrelevant idiosyncracies** specific to the training data (which **hurts** its generalization ability). We'll discuss this so-called "**over-fitting**" issue separately.