CSE 4502/5717 Big Data Analytics with Dr. Sanguthevar Rajasekaran
Lecture #16, 4/2/2018 Notes from Yenhsiang Lai

Machine learning is the task of inferring a function, e.g., $f: \mathbb{R}^m \rightarrow \mathbb{R}^n$. This inference has to be made using a series of *examples*. An example is nothing but a pair of the form $(\boldsymbol{x}, \boldsymbol{y})$, where $\boldsymbol{x} \in \mathbb{R}^m$ and $\boldsymbol{y} \in \mathbb{R}^n$ and $\boldsymbol{y}=f(\boldsymbol{x})$. The examples constitute the *training data* for machine learning.

Convention: Lowercase bold letters (such as $\boldsymbol{x}$ and $\boldsymbol{y}$) will be used to denote vectors and boldface capital letters (such as $\boldsymbol{A}$) will be used to denote matrices.

A simple example:

Assume that the function f is a degree k polynomial.

$f: \mathbb{R} \rightarrow \mathbb{R}$

$f = a_k x^k + a_{k-1} x^{k-1} + \ldots + a_1 x + a_0$

If we know the value of the polynomial at $(k+1)$ distinct points, we can uniquely determine $f(.)$ by interpolating.

In practice we may not know the form of the function and also there could be errors. In practice we guess the form of the function. Each such possible function will be called a *model* and characterized by some parameters.

We choose parameter values that will minimize the difference between the model outputs & the true function values.

MITCHELL (1997) :

A computer program is said to be learning to perform a set T of tasks from experience E under some performance measure P, if its performance with respect to the tasks in T improves with E.

There are two kinds of learning: **supervised** & **unsupervised**.
In **supervised learning** we are given a set of examples $(\boldsymbol{x}, \boldsymbol{y})$ and the goal is to infer the predicting conditional probability distribution $P(\boldsymbol{y} | \boldsymbol{x})$.
In **unsupervised learning** the goal is to predict a data generating distribution $P(\boldsymbol{x})$ after observing many random vectors $\boldsymbol{x}$ from this distribution.

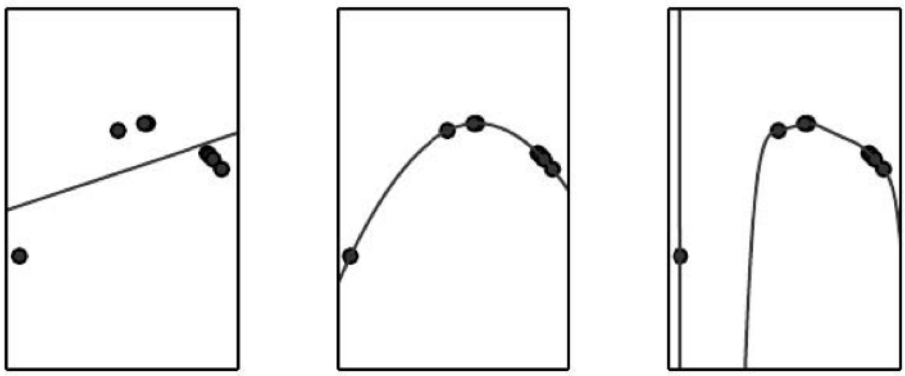
Capacity of a model:

A machine learning algorithm builds a model from the input training data. We can measure the accuracy of the model with respect to the training data. The corresponding error (i.e., the difference between the model and desired outputs) will be called the *training error*.

A machine learning algorithm will also be tested on data points previously unseen. These unseen data points used to test the algorithm will be referred to as the test data. We can define a corresponding *test error*.

A model is said to underfit if its training error is not low enough. A model is said to overfit if the difference between the training and test error is very large. In this case the model memorizes the properties of the training data closely. We can modify the underfitting and overfitting behavior of a learning algorithm by changing the capacity with a low capacity tends to underfit and a model with a high capacity tends to overfit.

(Goodfellow, et al. 2016) have generated data from a quadratic function and used three different models to fit the data. These three models were degree 1, degree 2, and degree 9 polynomials, respectively. The results they got are shown below:



The performance of a model is optimal when its capacity is close to the complexity of the function learned.

No free lunch theorem (WOLPERT 1996):

When averaged over all possible data generating distributions, each classifier has the same error rate on previously unseen data points. We can develop better

algorithms if we restrict the functions of interest and/or we have more information on the data generating distribution.

Example:

Linear regression: is one of the simple models we can think of. This model fits the data with a linear function.

Let $f = \mathbb{R}^n \rightarrow \mathbb{R}$ be any function, a linear regression model takes the form:
$f(\boldsymbol{x}) = w_1x_1 + w_2x_2 + \ldots + w_nx_n$. Here $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)^T$. Let $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^T$.

Input : m examples.

Output: $\boldsymbol{w} = (w_1, w_2, \ldots, w_n)^T$.

Let the examples be $E_1, E_2, \ldots E_m$, where $E_i = (x_i^1, x_i^2, \ldots, x_i^n, y_i)$, with $y_i = f(x_i^1, x_i^2, \ldots, x_i^n)$, $1 \leq i \leq m$.

Let $\hat{y}_i$ be the output of the model on input $\boldsymbol{x_i} = (x_i^1, x_i^2, \ldots, x_i^n)$.

Estimation error = $\frac{1}{m}\sum_{i=1}^{m}(\hat{y}_i - y_i)^2$. This is called the **mean squared error (MSE).**

$\hat{y}_i = w_1x_i^1 + w_2x_i^2 + \ldots + w_nx_i^n$.

Goal : Determine $\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ . \\ . \\ . \\ w_n \end{bmatrix}$ that minimizes the MSE.

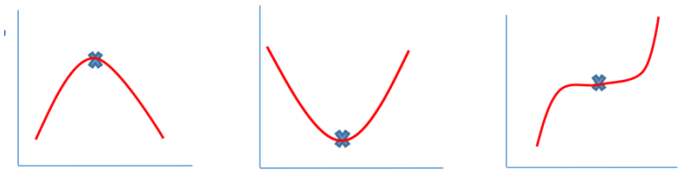
Gradient Descent:

Consider a function $f = \mathbb{R} \rightarrow \mathbb{R}$. Say we want to find $\operatorname{argmin}_x f(x)$.

$f(x+\varepsilon) \cong f(x) + \varepsilon f'(x)$, where $f'(x) = \frac{df}{dx}$.

Note: $f(x - \varepsilon \operatorname{sign} f'(x)) < f(x)$, for any small ε.

When $f'(x) = 0$, then x is a stationary point. A stationary point could be a (local) minimum, a (local) maximum, or a saddle point.



max min saddle point

Example:

$f(x) = 5x^2 - 10x + 7$

$\frac{df}{dx} = 10x - 10 = 0$, if $x = 1$.

$\frac{d^2f}{d^2x} = 10 \Rightarrow$ $x = 1$ is minimum.

Definition:

Let $\boldsymbol{y} = (y_1, y_2, \ldots, y_m)^T$ and $\boldsymbol{x} = (x_1, x_2, \ldots, x_n)$, then we define the gradient of $\boldsymbol{y}$ with respond to $\boldsymbol{x}$, denote as $\nabla_{\boldsymbol{x}}\boldsymbol{y}$ (or $\frac{\partial y}{\partial \boldsymbol{x}}$) $= \begin{bmatrix} \frac{\partial y_1}{\partial x_1} & \frac{\partial y_1}{\partial x_2} & \cdots & \frac{\partial y_1}{\partial x_n} \\ \frac{\partial y_2}{\partial x_1} & \frac{\partial y_2}{\partial x_2} & \cdots & \frac{\partial y_2}{\partial x_n} \\ \cdots & & & \\ \frac{\partial y_m}{\partial x_1} & \frac{\partial y_m}{\partial x_2} & \cdots & \frac{\partial y_m}{\partial x_n} \end{bmatrix}$

$\nabla_{\boldsymbol{x}}\boldsymbol{y}$ corresponds to a stationary point if each entry in the matrix is 0.

Some facts:

1. Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{y}$ is $m \times 1$, $\boldsymbol{A}$ is $m \times n$, and $\boldsymbol{x}$ is $n \times 1$. Then $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \boldsymbol{A}$.

   This is because $y_i = a_{i1}x_1 + a_{i2}x_2 + \ldots + a_{in}x_n$, $\frac{\partial y_i}{\partial x_j} = a_{ij}$, $\forall i,j$

2. Let $\boldsymbol{y} = \boldsymbol{A}\boldsymbol{x}$, where $\boldsymbol{y}$ is $m \times 1$, $\boldsymbol{A}$ is $m \times n$, and $\boldsymbol{x}$ is $n \times 1$, $\boldsymbol{A}$ is independent of $\boldsymbol{x}$ and $\boldsymbol{y}$ and $\boldsymbol{x}$ is a function of $\boldsymbol{z}$ then $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{z}} = \frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}} = \boldsymbol{A} \cdot \frac{\partial \boldsymbol{x}}{\partial \boldsymbol{z}}$.

3. Let $\alpha = \boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x}$, then $\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{y}^T\boldsymbol{A}$ and $\frac{\partial \alpha}{\partial \boldsymbol{y}} = \boldsymbol{x}^T\boldsymbol{A}^T$.

   Note: $\boldsymbol{y}^T\boldsymbol{A}\boldsymbol{x} = \boldsymbol{x}^T\boldsymbol{A}^T\boldsymbol{y}$.

4. If $\alpha = \boldsymbol{x}^T \boldsymbol{A}\, \boldsymbol{x}$, then $\frac{\partial \alpha}{\partial \boldsymbol{x}} = \boldsymbol{x}^T (\boldsymbol{A}+\boldsymbol{A}^T)$.

5. If $\alpha = \boldsymbol{x}^T \boldsymbol{A}\, \boldsymbol{x}$ and if $\boldsymbol{A}$ is symmetric, them $\frac{\partial \alpha}{\partial \boldsymbol{x}} = 2\boldsymbol{x}^T \boldsymbol{A}$.

Linear Regression:

We are required to minimize the MSE (i.e., we have to minimize $\frac{1}{m}\sum_{i=1}^{m}(\widehat{y_i} - y_i)^2$.

Let $\boldsymbol{X} = \begin{bmatrix} x_1^1\ x_1^2 \dots x_1^n \\ x_2^1\ x_2^2 \dots x_2^n \\ \vdots \\ x_m^1\ x_m^2 \dots x_m^n \end{bmatrix}$ $\boldsymbol{w} = \begin{bmatrix} w_1 \\ w_2 \\ \vdots \\ w_n \end{bmatrix}$. In this case, $\boldsymbol{Xw} = \begin{bmatrix} \widehat{y_1} \\ \widehat{y_2} \\ \vdots \\ \widehat{y_m} \end{bmatrix}$.

We have to minimize $\frac{1}{m}\| \boldsymbol{XW} - \boldsymbol{y}\|_2^2$. The plan is to set $\nabla_{\boldsymbol{w}}$ MSE = 0

$\Rightarrow \frac{1}{m}\nabla_{\boldsymbol{w}}\| \boldsymbol{XW} - \boldsymbol{y}\|_2^2 = 0$.

To be continued.