CSCI 4588/5588, Fall 2020
Machine Learning II
Study Guide for Test#1

Chapter 01

1. Write down the pseudo-code for the Genetic Algorithm.

Ans: Genetic Algorithm (GA) is a population-based optimization algorithm. The formation was inspired by natural evolution. A pseudo-code for GA is given below:

1. Form the initial population (usually random)
2. Compute the fitness to evaluate each chromosome (member of the population)
3. Select pairs to mate from best-ranked individuals and replenish population
   a. - Apply crossover operator
   b. - Apply mutation operator
4. Check for termination criteria, else go to step #2

Figure: Pseudocode for Genetic Algorithm

2. Write down the pseudo-code of the iterative hill-climbing algorithm.

Ans:

```
procedure iterated hillclimber
begin
  t ← 0
  repeat
    local ← FALSE
    select a current string v_c at random
    evaluate v_c
    repeat
      select 30 new strings in the neighborhood of v_c
        by flipping single bits of v_c
      select the string v_n from the set of new strings
        with the largest value of objective function f
      if f(v_c) < f(v_n)
        then v_c ← v_n
        else local ← TRUE
    until local
    t ← t + 1
  until t = MAX
end
```

3. Write down the pseudo-code of the simulated annealing algorithm.

Ans:

```
procedure simulated annealing
begin
   t ← 0
   initialize temperature T
   select a current string v_c at random
   evaluate v_c
   repeat
      repeat
         select a new string v_n
            in the neighborhood of v_c
            by flipping a single bit of v_c
         if f(v_c) < f(v_n)
            then v_c ← v_n
            else if random[0, 1) < exp{(f(v_n) − f(v_c))/T}
                 then v_c ← v_n
      until (termination-condition)
      T ← g(T, t)
      t ← t + 1
   until (stop-criterion)
end
```

4. Name ten non-deterministic algorithms (Excluding Genetic Algorithms, Hill-Climbing, Simulated Annealing).

Ans:

Particle Swarm Optimization (PSO),
Artificial Colony Optimization (ACO),
Artificial Bee Colony (ABC),
Glowworm Swarm Optimization (GSO),
Cuckoo Search Algorithm (CSA),
Firefly Algorithm (FA),
Bat Algorithm (BA),
Monkey Algorithm (MA),
Krill Herd Algorithm (KHA),
Wind Driven Optimization (WDO),
Social Spider Algorithm (SSA),
Artificial Immune Systems,
Conformational Space Annealing, …

Chapter 02

1. What is supervised learning?

2. What is the overfitting problem?

3. What are the differences between 'regression' and 'classification' problems? [~Chapter #2, Page 8]

4. Describe Newton's method for deriving the equation for finding the minimum (or maximum) of a given equation.

Ans:
Say, we have an equation $f(x) = 0$

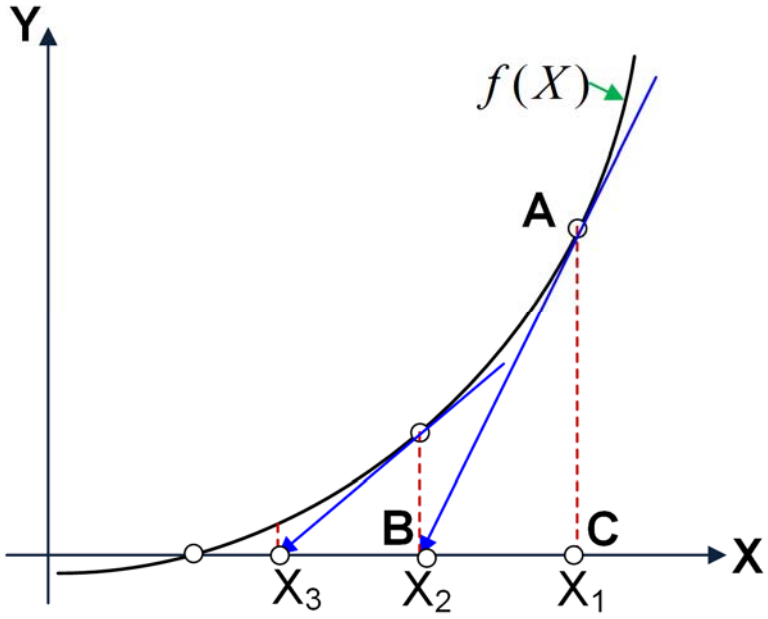


Figure: Newton's method is used for finding the solution of an equation.

For the solution of the equation (i.e., to find for what value of x, $f(x) = 0$), assume our initial point is x_1 which intersect x-axis at C and $f(x)$ at A (see Figure above). Also, assume that the tangent at A intersect x-axis at B where the value of x is x_2. From ΔABC and the definition of the slope of an equation, we can write:

$$f'(x_1) = \frac{f(x_1) - 0}{x_1 - x_2} \qquad (i)$$

$$x_2 = x_1 - \frac{f(x_1)}{f'(x_1)} \qquad (ii)$$

In general, we can write,

$$x_{t+1} = x_t - \frac{f(x_t)}{f'(x_t)} \qquad (iii)$$

Equation (*iii*) can be iteratively used to get the solution of the equation.

Now, for a minimization problem, if minimum exists then, we need to find, the value of x for which $f'(x) = 0$.

We can think of going for the solution of the equation, $f'(x) = 0$ using (*iii*).

Therefore, we can similarly write,

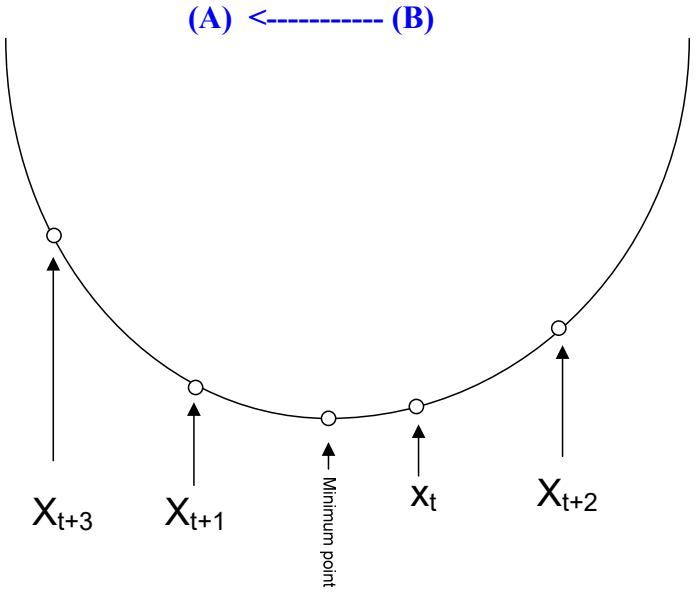
$$x_{t+1} = x_t - \frac{f'(x_t)}{f''(x_t)} \qquad (iv)$$

Now, we can use equation (*iv*) to find the minimum (or, maximum) of equation $f(x) = 0$.

5. (a) What is the gradient descent equation? What is the α in that equation used for? (b) How the overshooting problem can occur with a gradient descent approach? (c) How does the gradient ascent equation differ from descent equation?

Ans: (a) [do it by yourself, check lecture/note]

Ans: (b) Hints:
If we set the value of α (alpha, the learning rate), to a higher value, then overshooting might occur. How?



Say we are after the minimum point.
Assume, we are at X(t) now.

We set α a very high value and compute, x(t+1) as:

$$x_{t+1} = x_t - \alpha \ \nabla f(x_t)$$

Note that the α is constant – has a fixed value or it does not change in the iteration but remember slope $\nabla f(x_i)$ dependent on the location of x_i and it changes.

Now, x(t+1) surpasses the minimum point when moving from **(B) to (A) direction** for the minimum point for a very higher deduction (amount: $\alpha \ \nabla f(x_t)$) from x(t).

From the figure, we see a distance of x(t+1) from the minimum point is higher than the an of x(t) from a minimum point. So, it is obvious that:

$$|\nabla f(x_{t+1})| > |\nabla f(x_t)|$$

[in this case, $\nabla f(x_t)$ is +ve and $\nabla f(x_{t+1})$ is -ve]

$\alpha |\nabla f(x_{t+1})| > \alpha |\nabla f(x_t)|$ will be true.

Based on this information and the figure, we can say that the next point x(t+2) would be behind x(t) as we will apply:

$$x_{t+2} = x_{t+1} - \alpha \ \nabla f(x_{t+1})$$

Again $\alpha |\nabla f(x_{t+2})| > \alpha |\nabla f(x_{t+1})| > \alpha |\nabla f(x_t)|$ will be true, so the next point x(t+3) will be behind x(t+1), etc.

So, we see instead of converging, it is diverging in each iteration => overshooting.

Ans: (c) Hints:

Gradient descent (climbing down) is given by: $x_{t+1} = x_t - \alpha \ \nabla f(x_t)$

Gradient ascent (climbing up) is given by: $x_{t+1} = x_t + \alpha \ \nabla f(x_t)$

(A) <---------- (B) **(A) <---------- (B): direction**

(a) (b)

Figure: Curve has (a) minimum point and (b) maximum point.

For curve (a) when the iteration goes from **(B) to (A)** to get the minimum point the slope (i.e., ∇f or, $f'(x)$) remains +ve. And we set the α (alpha; the learning rate) as a positive fixed number. So to reduce the x_t in each iteration to move from **(B) to (A)** the term '$\alpha \nabla f(x_t)$' must be deducted.

Now, for the curve (b), we cannot find the minimum point (the minimum point does not exist). Instead, we have a maximum point. Using the equation, iteratively if we are moving from **(B) to (A)**, the slope in the case will be –ve, and α is positive, so the term $\alpha \nabla f(x_t)$ is now negative. So, we need to have a +ve sign in front of the term to keep it negative so, that the x_t is deducted in each iteration to move it from **(B) to (A) direction**.

10. The equation $RSS(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$ is a quadratic function of parameters β; therefore, its minimum always exists. How can we say so? Explain for the given equation $RSS(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$, where, RSS implies a residual sum of squares. [See page ~ 12-13, Chapter #2].

11. We can write the RSS equation, $RSS(\beta) = \sum_{i=1}^{N} (y_i - x_i^T \beta)^2$, in vector form as: $\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$, where X is an *N* × *p* matrix with each row is an input vector, and y is an *N*-vector of the outputs in the training set. Differentiating w.r.t. β, we get the *normal equations*: $\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$.

Show the steps for getting $\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$ from $\text{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta)$.

Ans:

$$
\begin{aligned}
RSS(\beta) &= (\mathbf{y} - \mathbf{X}\beta)^T (\mathbf{y} - \mathbf{X}\beta) \\
&= [\mathbf{y}^T - (\mathbf{X}\beta)^T](\mathbf{y} - \mathbf{X}\beta) \qquad [\because (A \pm B)^T = A^T \pm B^T,\ (AB)^T = B^T A^T] \\
&= (\mathbf{y}^T - \beta^T \mathbf{X}^T)(\mathbf{y} - \mathbf{X}\beta) \\
&= \mathbf{y}^T\mathbf{y} - \beta^T \mathbf{X}^T \mathbf{y} - \mathbf{y}^T \mathbf{X}\beta + \beta^T \mathbf{X}^T \mathbf{X}\beta \qquad \left[\begin{array}{l} \because a^T b = b^T a \\ \therefore \mathbf{y}^T \mathbf{X}\beta = (\mathbf{X}\beta)^T \mathbf{y} = \beta^T \mathbf{X}^T \mathbf{y} \end{array}\right] \\
&= \mathbf{y}^T\mathbf{y} - \beta^T \mathbf{X}^T \mathbf{y} - \beta^T \mathbf{X}^T \mathbf{y} + \beta^T \mathbf{X}^T \mathbf{X}\beta \\
&= \mathbf{y}^T\mathbf{y} - 2\beta^T \mathbf{X}^T \mathbf{y} + \beta^T \mathbf{X}^T \mathbf{X}\beta
\end{aligned}
$$

Therefore, we have

$$RSS(\beta) = \mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{X}\beta$$

Now, differentiating w.r.t. β and equating it to zero, we get:

$$\frac{\partial}{\partial\beta}[\mathbf{y}^T\mathbf{y} - 2\beta^T\mathbf{X}^T\mathbf{y} + \beta^T\mathbf{X}^T\mathbf{X}\beta] = 0$$

$$\Rightarrow \frac{\partial}{\partial\beta}(\mathbf{y}^T\mathbf{y}) - 2\frac{\partial}{\partial\beta}(\beta^T)\mathbf{X}^T\mathbf{y} + \frac{\partial}{\partial\beta}(\beta^T\mathbf{X}^T\mathbf{X}\beta) = 0$$

$$\Rightarrow 0 - 2\mathbf{X}^T\mathbf{y} + \frac{\partial}{\partial\beta}(\beta^T)\mathbf{X}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X}\frac{\partial}{\partial\beta}(\beta) = 0$$

$$\Rightarrow 0 - 2\mathbf{X}^T\mathbf{y} + \mathbf{X}^T\mathbf{X}\beta + \beta^T\mathbf{X}^T\mathbf{X} = 0 \qquad \left[\begin{array}{l} \because \beta^T\mathbf{X}^T\mathbf{X} = \beta^T(\mathbf{X}^T\mathbf{X}) \\ = (\mathbf{X}^T\mathbf{X})^T\beta = (\mathbf{X}^T)(\mathbf{X}^T)^T\beta \\ = \mathbf{X}^T\mathbf{X}\beta \end{array}\right]$$

$$\Rightarrow -2\mathbf{X}^T\mathbf{y} + 2\mathbf{X}^T\mathbf{X}\beta = 0$$

$$\Rightarrow -2\mathbf{X}^T(\mathbf{y} - \mathbf{X}\beta) = 0$$

$$\therefore \boldsymbol{X}^T(\boldsymbol{y} - \boldsymbol{X}\beta) = 0$$

12. The $RSS(\beta) = \sum_{i=1}^{N}(y_i - x_i^T\beta)^2$ equation can be expressed in the vector form as: $\mathrm{RSS}(\beta) = (\mathbf{y} - \mathbf{X}\beta)^T(\mathbf{y} - \mathbf{X}\beta)$, where X is an *N* × *p* matrix with each row is an input vector, and y is an *N*-vector of the outputs in the training set. (a) Find the vector form of the regularized version of the RSS, i.e., $RSS(\beta) = \sum_{i=1}^{N}\left[(\hat{y}(x_i,\beta) - y_i)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right]$ and (b) show the steps to derive, $\hat{\beta} = (\mathbf{X}^T\mathbf{X} + \lambda M_\lambda)^{-1}\mathbf{X}^T\mathbf{y}$, Here, M_λ is a (*p*+1) by (*p*+1) identity matrix except, $M_\lambda(1,1) = 0$.

Ans:

(a) The vector form of the equation is $\mathrm{RSS}(\beta) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$

(b) Here, $\mathrm{RSS}(\beta) = (y - X\beta)^T(y - X\beta) + \lambda\beta^T\beta$

$= (y^T - \beta^T X^T)(y - X\beta) + \lambda\beta^T\beta \quad [\because (\mathrm{A} \pm \mathrm{B})^{\mathrm{T}} = \mathrm{A}^{\mathrm{T}} \pm \mathrm{B}^{\mathrm{T}}, (\mathrm{AB})^{\mathrm{T}} = \mathrm{B}^{\mathrm{T}}\mathrm{A}^{\mathrm{T}}]$

$= y^T y - \beta^T X^T y - y^T X\beta + \beta^T X^T X\beta + \lambda\beta^T\beta \quad [\because a^T b = b^T a, \therefore y^T X\beta = (X\beta)^T y = \beta^T X^T y]$

$= y^T y - \beta^T X^T y - \beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta$

$= y^T y - 2\beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta$

Therefore, we get the expanded form:

$$RSS(\beta) = y^T y - 2\beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta$$

Now, differentiating $\text{RSS}(\beta) = y^T y - 2\beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta$ with respect to β and equating it to zero, we get:

$$\frac{\partial}{\partial\beta}(y^T y - 2\beta^T X^T y + \beta^T X^T X\beta + \lambda\beta^T\beta) = 0$$

$$=> 0 - 2X^T y + \frac{\partial}{\partial\beta}(\beta^T)X^T X\beta + \beta^T X^T X\frac{\partial}{\partial\beta}(\beta) + \frac{\partial}{\partial\beta}(\lambda\beta^T I\beta) = 0$$

[Here, M_λ is the indetify martix, and we write $\lambda\beta^T\beta => \lambda\beta^T\text{M}_\lambda\beta$]

$$=> 0 - 2X^T y + X^T X\beta + \beta^T X^T X + \lambda\text{M}_\lambda\beta + \lambda\beta^T\text{M}_\lambda = 0$$

$$[\because \beta^T X^T X = \beta^T(X^T X) = (X^T X)^T\beta = X^T X\beta)]$$

$$=> -2X^T y + 2X^T X\beta + 2\lambda\text{M}_\lambda\beta = 0$$

$$[\because \beta^T\text{M}_\lambda = \text{M}_\lambda{}^T\beta = \text{M}_\lambda\beta]$$

$$=> -2X^T y + 2\beta(X^T X + \lambda\text{M}_\lambda) = 0$$

$$=> X^T y = \beta(X^T X + \lambda\,\text{M}_\lambda)$$

$$=> \beta = (X^T X + \lambda\,\text{M}_\lambda)^T X^T y\,.$$

13. With regularization, the RSS equation with minimization target can be expressed as:

$$\min_{RSS_\lambda(\beta)} = \sum_{i=1}^{N}\left[\left(\hat{y}(x_i,\beta) - y_i\right)^2 + \lambda\sum_{j=1}^{p}\beta_j^2\right],$$

where λ is the regularization parameter. The derived gradient descent approach from the above equation can be expressed as:

$$\beta_j(t+1) = \beta_j(t) + \frac{2\alpha}{N}\left[\sum_{i=1}^{N}\left(y(i) - x^T(i)\,\beta\right).x(i)_j - \lambda\beta(t)_j\right]$$

From the equation of the gradient descent approach with regularization, show that the parameter β is shrinking.

Ans:

For the given equation, $\beta_j(t+1) = \beta_j(t) + \frac{2\alpha}{N}\left[\sum_{i=1}^{N}\left(y(i) - x^T(i)\,\beta\right).x(i)_j - \lambda\beta(t)_j\right]$

we can rewrite it by rearranging terms as:

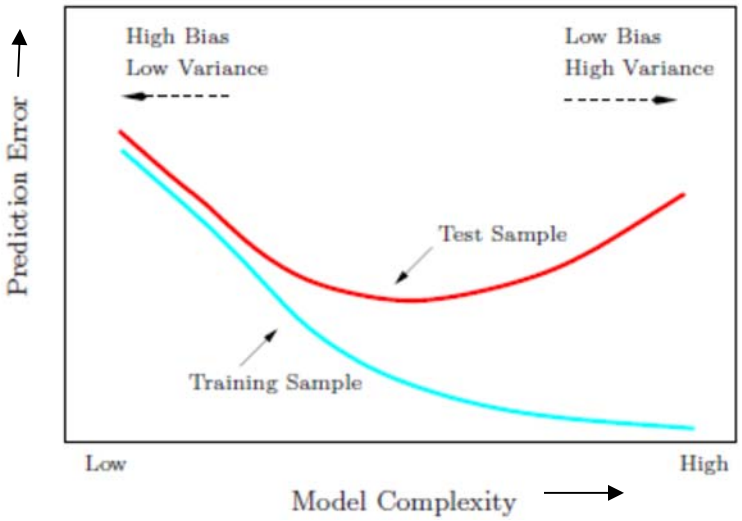
$$\beta_j(t+1) = \beta_j(t) - \frac{2\alpha}{N}.\lambda\beta(t)_j + \frac{2\alpha}{N}\left[\sum_{i=1}^{N}\left(y(i) - x^T(i)\,\beta\right).x(i)_j\right]$$

$$\Rightarrow \beta_j(t+1) = \beta_j(t)(1-\frac{2\alpha\lambda}{N}) + \frac{2\alpha}{N}\left[\sum_{i=1}^{N} \left(y(i) - x^T(i)\,\beta\right).x(i)_j\right]$$

We see the $\beta_j(t)$ is actually shrinking due to the term $(1-\frac{2\alpha\lambda}{N})$ which is < 1 because α, λ and N are positive quantities.

14. (a) Draw a figure to show the relationship among *Prediction Error*, *Model Complexity*, *Bias*, *Variance*, *Training dataset,* and *Test dataset*, (b) What is the relation between, the Residual Sum of Squares (RSS) and the Mean Squared Error (MSE), (c) What is the relationship among MSE, Bias Error, and Variance Error?

Ans:
(a)



(b) The relation between the Residual Sum of Squares (RSS) and the Mean Squared Error (MSE) is given by the following equation:

$$MSE = \frac{1}{N} RSS$$

Here, N is the number of actual and predicted data pairs (pair of actual and its corresponding predicted data points) in RSS.

(c) The relationship among MSE, Bias-Error, and Variance-Error are given by the following equation:

$$MSE = (\text{Bias-Error})^2 + \text{Variance-Error}$$

(15) Write down the steps for hold-out cross-validation or, simple cross-validation for model selection.

Ans:

1. Randomly split data (D): D_{train} and $D_{holdout}$ (Say, 25-30% of the data).
2. For $\forall i$, train model M_i using D_{train} and get the corresponding P_i (i^{th} predictor).
3. For $\forall i$, compute the error E_i of P_i using $D_{holdout}$.
4. Pick the predictor where the error is the lowest.

(16) Write down the steps for k-fold cross-validation for model selection.

Ans:

1. Randomly split data (D) into *k* disjoint subsets as: $D_1, D_2, ..., D_k$.
2. For $\forall i$, model M_i is evaluated as:
   For j =1 to k:
   Train the M_i using data: $(D - D_j)$ and get the predictor P_{ij}
   Test P_{ij} using D_j and get the error E_{ij}.
   EndFor j

   E_i = average of E_{ij} for $\forall j$, which is the generalized error of M_i.
3. Pick the best model M_i having the lowest generalized error E_i.
4. Retrain $M_{i=best}$ using full dataset D.

(17) Write down the steps for predicting the best values for the regularization parameter λ.

Ans:

1. Take a range of values with regular intervals: {0, …, R} of real or integer numbers for λ.
2. Evaluate the performance for each of the value of λ using cross-validation approach for example.
3. Pick the best value of λ for which the error was lowest.

--- x ---