

Machine Learning End Term Exam

Ahmad Shayaan IMT2014004
H Vijaya Sharvani IMT2014022
Indu Ilanchezian IMT2014024

December 15, 2017

Question 2

To derive the solution to the modified linear regression and to show that it leads to the generalized form of ridge regression.

Solution:-

Given the attribute $x_i = \hat{x}_i + \epsilon_i$, where the \hat{x}_i is the true measurement and ϵ_i is the zero mean vector with covariance matrix $\sigma^2 I$.

Modified loss function

$$W^* = \underset{W}{\operatorname{argmin}} E_{\epsilon} \sum_{i=1}^n (y_i - W^T(\hat{x}_i + \epsilon_i))^2$$

Where W is the transformation vector.

$$W^* = \underset{W}{\operatorname{argmin}} E_{\epsilon} \|Y - (X + \epsilon)W\|_2^2 \quad (1)$$

Where

$$Y = \begin{bmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{bmatrix}$$

$$X = \begin{bmatrix} \hat{x}_1^T \\ \hat{x}_2^T \\ \vdots \\ \hat{x}_n^T \end{bmatrix}$$

$$\epsilon = \begin{bmatrix} \epsilon_1^T \\ \epsilon_2^T \\ \vdots \\ \epsilon_n^T \end{bmatrix}$$

Expanding right hand side of equation (1).

$$\begin{aligned} E_\epsilon \|Y - (X + \epsilon)W\|_2^2 &= E_\epsilon \left[(Y - (X + \epsilon)W)^T (Y - (X + \epsilon)W) \right] \\ &= E_\epsilon \left[Y^T Y + W^T (X + \epsilon)^T (X + \epsilon) W - 2W^T (X + \epsilon)^T Y \right] \end{aligned} \quad (2)$$

To minimize the equation we will differentiate equation (2) with respect to W.

$$\frac{\partial E_\epsilon \left[Y^T Y + W^T (X + \epsilon)^T (X + \epsilon) W - 2W^T (X + \epsilon)^T Y \right]}{\partial W} = 0$$

We know that $\frac{\partial E(f(x))}{\partial} = E \frac{\partial f(x)}{\partial x}$.

$$E_\epsilon \left[\frac{\partial Y^T Y}{\partial W} + \frac{\partial W^T (X + \epsilon)^T (X + \epsilon) W}{\partial W} - 2 \frac{\partial W^T (X + \epsilon)^T Y}{\partial W} \right] = 0$$

$$E_\epsilon [2(X + \epsilon)^T (X + \epsilon) W - 2(X + \epsilon)^T Y] = 0$$

$$2E_\epsilon [(X + \epsilon)^T (X + \epsilon) W] - 2E_\epsilon [(X + \epsilon)^T Y] = 0$$

$$E_\epsilon [(X^T X + \epsilon^T \epsilon + 2\epsilon^T X) W] = E_\epsilon [(X + \epsilon)^T Y]$$

$$E_\epsilon (X^T X W) + E_\epsilon (\epsilon^T \epsilon W) + 2E_\epsilon (\epsilon^T X W) = E_\epsilon (X^T Y) + E_\epsilon (\epsilon^T Y)$$

We know that $E(AB) = E(A)E(B)$ if A and B are independent variables and $E_f(h(x)) = \int_{-\infty}^{\infty} h(x)f(x)dx$.

$$\sum_{i=1}^n X^T X W P(\epsilon_i) + E_\epsilon (\epsilon \epsilon^T) E_\epsilon (W) + 2E_\epsilon (X) E_\epsilon (\epsilon) = \sum_{i=1}^n X^T Y P(\epsilon_i) + E_\epsilon (Y) E_\epsilon (\epsilon)$$

We know that the noise is a zero mean Gaussian noise therefore $E_\epsilon (\epsilon) = 0$

$$(X^T X + \sigma^2 I) W = X^T Y$$

$$W = (X^T X + \sigma^2 I)^{-1} X^T Y$$

Therefore the solution of the minimization is

$$W^* = (X^T X + \sigma^2 I)^{-1} X^T Y$$

This solution is same as the solution for Ridge regression.

$$W^* = (X^T X + \lambda I)^{-1} X^T Y$$

Question 3

$VC(\mathcal{H})$ is the maximum cardinality of any set of instances that can be shattered by \mathcal{H} . We say that \mathcal{H} shatters a set of points if and only if it can assign any possible labeling to those points.

1. We should show that the VC dimension $d_{\mathcal{H}}$ of any finite hypothesis space \mathcal{H} is at most $\log_2 |\mathcal{H}|$.

Proof:

For any set of distinct points S of size m , there are 2^m distinct ways of labeling those points. This means that for \mathcal{H} to shatter S it must contain at least 2^m distinct hypotheses. This tells us that if the VC dimension of \mathcal{H} is m then we must have 2^m hypotheses, i.e. $2^m \leq |\mathcal{H}|$ or equivalently that $m = VC(\mathcal{H}) \leq \log_2 |\mathcal{H}|$.

2. We should show that the bound for VC dimension is tight and for any $d > 1$, there exists a hypothesis class \mathcal{H} such that $d = d_{\mathcal{H}}$.

Proof:

Consider a domain with n binary features and binary class labels. Let \mathcal{H} be the hypothesis space that contains all decision trees over those features that have depth no greater than d_e . (The depth of a decision tree is the depth of the deepest leaf node.)

First note that any tree in \mathcal{H} can be represented by a tree of exactly depth d_e in \mathcal{H} . So we will restrict our attention to trees of exactly depth d_e . All of these trees have 2^{d_e} leaf nodes. Also note that there are a total of 2^n examples in our instance space, which gives us an immediate upper bound on the VC-dimension of \mathcal{H} , i.e. $VC(\mathcal{H}) \leq 2^n$.

To get a lower bound let S contain the set of all possible 2^n instances. Since we have that $d_e \geq n$ it is straightforward to create a tree of depth n with a leaf node for each example and furthermore we can label the leaf nodes in all possible ways. This shows that we can shatter the set S with \mathcal{H} , which implies that $VC(\mathcal{H}) \geq 2^n$. Combining the upper and lower bound tell us that $VC(\mathcal{H}) = 2^n$, i.e. $d_{\mathcal{H}} = 2^n$. Hence, we have showed a tight bound on the VC dimension of hypothesis space \mathcal{H} .

Now we will show that for any $d > 1$, there exists a hypothesis class \mathcal{H} such that $d = d_{\mathcal{H}}$.

Take a set of size d , $C = \{e_1, e_2, \dots, e_d\}$ such that $\{e_i; i \in [d]\}$ is the standard basis in R^d . To prove that C shatters \mathcal{H} , it suffices to show that $|\mathcal{H}_C| = 2^d$. The hypothesis on the set C is given as,

$$\mathcal{H}_C = \{h(c), h \in HS_d\} = \{h(e_1), h(e_2), \dots, h(e_d), h \in HS_d\}$$

For a particular $\omega^T = (\omega_1, \omega_2, \dots, \omega_d)$,

$$\begin{aligned} h(c) &= (\langle \omega, c \rangle, c \in C) \\ &= (\langle \omega_1, e_1 \rangle, \langle \omega_2, e_2 \rangle, \dots, \langle \omega_d, e_d \rangle) \\ &= (\omega_1, \omega_2, \dots, \omega_d) \\ &= \omega \end{aligned}$$

Since all possible combinations 2^d be chosen on ω .

$$\begin{aligned} \implies |\mathcal{H}_C| &= 2^d \\ \implies VCdim(HS_d) &\geq d \end{aligned}$$

Let us take a arbitrary set C of size d + 1.

$$C = \{x_1, x_2, \dots, x_{d+1}\}, x_i \in R^d$$

Since, x'_i s are coming from d dimensional space, $\{x_i, i \in [d+1]\}$ are linearly dependent,

$$\implies \exists a_1, a_2, \dots, a_{d+1} \text{ s.t. } \sum_{i=1}^{d+1} a_i x_i = 0$$

Let $I = \{i, a_i > 0\}$ and $J = \{j, a_j < 0\}$

$$\implies \sum_{i \in I} a_i x_i = - \sum_{j \in J} a_j x_j = \sum_{j \in J} |a_j| x_j$$

Suppose C is shattered by \mathcal{H} .

$$\text{Claim : } \exists \omega \text{ s.t } \langle \omega, x_i \rangle > 0 \forall i \in I \text{ \& } \langle \omega, x_j \rangle < 0, \forall j \in J$$

$$\begin{aligned} \implies 0 &< \sum_{i \in I} a_i \langle \omega, x_i \rangle \\ &= \sum_{i \in I} \langle \omega, a_i x_i \rangle \\ &= \sum_{j \in J} \langle \omega, |a_j| x_j \rangle \\ &= \sum_{j \in J} |a_j| \langle \omega, x_j \rangle < 0, \text{ which is a contradiction} \end{aligned}$$

Therefore $|\mathcal{H}_C| < 2^{d+1}$ for any arbitrary set of size d + 1. So, the VC-dimension of the class of homogeneous halfspace in R^d is d.

Question 4

Algorithm 1 Building Decision tree

```

1: procedure BUILDDECISIONTREE(Dataset D, Target_Attributes, Attributes)
2:   T =  $\phi$  ▷ Initializing a tree
3:   if All Target attributes of one type then
4:     return a node with single label
5:   else if Attributes =  $\phi$  then
6:     return single node tree i.e. root with label = most common value of
7:       the target attribute in the dataset
8:   else
9:     Attributes* = GETCHISQUARESCORE(Attributes)
10:    A = GETINFOGAIN(Attributes*, Target_Attributes)
11:    for all possible values of A do
12:      if A has missing values then
13:        n = Total number of data points in A
14:        n* = Number of non-missing values
15:         $\mu_i = \frac{n_i^*}{n^*} \dots \frac{n_{m_i}^*}{n^*}$ 
16:        M = GENMISSINGVALUE( $\mu_i(n - n^*)$ ) ▷ Function generate
        missing values with certain probabilities
17:        D[A].append(M) ▷ Adding missing values of attribute A
18:        missing_probab.append( $\mu_i$ ) ▷ Store the missing value
        probabilities
19:        subset = The set of data points with value  $v_i$  for A
20:        T.addNode[ BuildDecisionTree(subset, Target_Attributes, Attributes-
        A) ]
21:   return (T, missing_probab)
22: function GETCHISQUARESCORE(Attributes)
23:   l = [ ] ▷ Empty list initialization
24:   for A in Attributes do
25:      $\mu_i = \frac{n_i}{n}$  ▷  $n_i$ : number of data points with  $i^{th}$  label
26:      $n_{ij}$  = number of points with label i in partition j
27:      $e_{ij} = \mu_i \sum_i n_{ij}$ 
28:      $score = \sum_i \frac{(n_{ij} - e_{ij})^2}{e_{ij}}$ 
29:     if score  $\geq$  threshold then ▷ Correlation for partition j
30:       l.append(A)
31:   return l ▷ list containing attributes with score greater than threshold
32: function GETINFOGAIN(Attributes*, Target_Attributes)
33:   Target_Entropy =  $-\sum_i P(y = i) \log(P(y = i))$  ▷ Entropy of the target
   variable
34:   for all i in Attributes* do
▷  $P(y = k | x_i = v_{ij}) = \frac{\theta_{ijk} P(y=k)}{P(x_i=v_{ij})}$ 
35:     Attributes_Entropy =  $-\sum_{ijk} \frac{\theta_{ijk} P(y=k)}{P(x_i=v_{ij})} \log \left[ \frac{\theta_{ijk} P(y=k)}{P(x_i=v_{ij})} \right]$ 
36:     gain = 0
37:     gain = Target_Entropy - Attributes_Entropy
38:   return Attribute with max information Gain

```

Algorithm 2 Prediction using decision tree

```
1: procedure TESTTREE(Tree T, Datapoint D, missing_probab)
2:                                     ▷ Node is a list that stores all the leaf nodes
3:   if T.node == Leaf or D ==  $\phi$  then
4:     return T.node
5:   for all attributes A in D do                                     ▷ A is an attribute of D
6:     if A is not in T.node then
7:       continue
8:     else
9:       if value(A) !=  $\phi$  then
10:        d = T.checknode(value(A))
11:                                     ▷ d is the decision taken at that node
12:        Node.append[TESTTREE(T.takepath(d), D-A, missing_probab) ]
13:                                     ▷ Take path along branch taken according to decision d
14:      else if value(A) ==  $\phi$  then
15:        for All probabilities p in missing_probab do
16:          ▷ probabilities stored for missing values of attributes at
the time of testing
17:          Push decision tree in each branch of the node with proba-
bility  $\mu_i$ 
18:          Node.append(all the returned leaf nodes)
19:      if Node.size == 1 then
20:        return Probability 1 for that class and zero for other
21:      else
22:        S = Sum of probabilities for each class
23:        ▷ S is a list that contains class membership probabilities
24:      return S
```

Question 5

Let $\vec{x}_1, \vec{x}_2, \dots, \vec{x}_n$ be the feature vectors of n data points in the original feature space. Let ϕ be the feature transformation function. Then, $\phi(\vec{x}_1), \phi(\vec{x}_2), \dots, \phi(\vec{x}_n)$ are the feature vectors in the transformed feature space.

Let K be the kernel function such that:

$$K(i, j) = \phi(x_i)^T \phi(x_j)$$

The center of mass, $\vec{\mu}$, in the feature space can be defined as the average of the vectors in the transformed feature space.

$$\vec{\mu} = \frac{1}{n} \sum_{i=1}^n \phi(\vec{x}_i)$$

Consider:

$$\begin{aligned}
||\mu||^2 &= \mu^T \mu \\
&= \mu^T \frac{1}{n} \sum_{i=1}^n \phi(\vec{x}_i) \\
&= \frac{1}{n} \sum_{j=1}^n \phi(\vec{x}_j)^T \frac{1}{n} \sum_{i=1}^n \phi(\vec{x}_i) \\
&= \frac{1}{n^2} \sum_{i,j} \phi(\vec{x}_j)^T \phi(\vec{x}_i) \\
&= \frac{1}{n^2} \sum_{i,j} K(i, j)
\end{aligned}$$

Average of the squared Euclidean distances from μ to each $\phi(x)$

The squared euclidean distance of a single feature vector in the transformed space from the center of mass $\vec{\mu}$ can be expressed as follows:

$$\begin{aligned}
||\phi(\vec{x}_i) - \vec{\mu}||^2 &= (\phi(\vec{x}_i) - \vec{\mu})^T (\phi(\vec{x}_i) - \vec{\mu}) \\
&= \phi(\vec{x}_i)^T \phi(\vec{x}_i) - 2\phi(\vec{x}_i)^T \vec{\mu} + ||\vec{\mu}||^2 \\
&= K(i, i) - \frac{2}{n} \phi(\vec{x}_i)^T \sum_{j=1}^n \phi(\vec{x}_j) + ||\vec{\mu}||^2 \\
&= K(i, i) - \frac{2}{n} \sum_{j=1}^n \phi(\vec{x}_i)^T \phi(\vec{x}_j) + ||\vec{\mu}||^2 \\
&= K(i, i) - \frac{2}{n} \sum_{j=1}^n K(i, j) + \frac{1}{n^2} \sum_{r,s} K(r, s)
\end{aligned}$$

The average of the euclidean distances of all the points from the center of mass can be written as:

$$\begin{aligned}
\frac{1}{n} \sum_{i=1}^n ||\phi(\vec{x}_i) - \vec{\mu}||^2 &= \frac{1}{n} \left(\sum_{i=1}^n \left(K(i, i) - \frac{2}{n} \sum_{j=1}^n K(i, j) + \frac{1}{n^2} \sum_{r,s} K(r, s) \right) \right) \\
&= \frac{1}{n} \left(\sum_{i=1}^n K(i, i) - \frac{2}{n} \sum_{i,j} K(i, j) + \frac{n}{n^2} \sum_{r,s} K(r, s) \right) \\
&= \frac{1}{n} \left(\sum_{i=1}^n K(i, i) - \frac{1}{n} \sum_{i,j} K(i, j) \right)
\end{aligned}$$

Thus, the average of euclidean distances from the center of mass $\vec{\mu}$ to each $\phi(x)$ can be expressed in terms of the kernel function K .