YSU Statistical ML, Fall 2019

Lecture 03

Michael Poghosyan
YSU, AUA
michael@ysu.am, mpoghosyan@aua.am

19 November 2019

Contents

Last Lecture Recap: Least Squares Regression Problem

Recall that in the case of LS Regression Problem,

- $\mathcal{X}$ is arbitrary, $\mathcal{Y} \subset \mathbb{R}$;

Last Lecture Recap: Least Squares Regression Problem

Recall that in the case of LS Regression Problem,

- $\mathcal{X}$ is arbitrary, $\mathcal{Y} \subset \mathbb{R}$;
- Loss Function is the Quadratic Loss: $\ell(y_1, y_2) = (y_1 - y_2)^2$

Last Lecture Recap: Least Squares Regression Problem

Recall that in the case of LS Regression Problem,

- $\mathcal{X}$ is arbitrary, $\mathcal{Y} \subset \mathbb{R}$;
- Loss Function is the Quadratic Loss: $\ell(y_1, y_2) = (y_1 - y_2)^2$

And, if g is any Predictor, then

$$Risk(g) = \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big) = \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big).$$

Last Lecture Recap: Least Squares Regression Problem

Recall that in the case of LS Regression Problem,

- $\mathcal{X}$ is arbitrary, $\mathcal{Y} \subset \mathbb{R}$;
- Loss Function is the Quadratic Loss: $\ell(y_1, y_2) = (y_1 - y_2)^2$

And, if g is any Predictor, then

$$Risk(g) = \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big) = \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big).$$

The Problem to find the Bayes Predictor in this case is:

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big).$$

Last Lecture Recap: Least Squares Regression Problem

Recall that in the case of LS Regression Problem,

- $\mathcal{X}$ is arbitrary, $\mathcal{Y} \subset \mathbb{R}$;
- Loss Function is the Quadratic Loss: $\ell(y_1, y_2) = (y_1 - y_2)^2$

And, if g is any Predictor, then

$$Risk(g) = \mathbb{E}\Big(\ell(Y, g(\mathbf{X}))\Big) = \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big).$$

The Problem to find the Bayes Predictor in this case is:

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big).$$

Now, let us find a Bayes Predictor in this case.

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$.

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation.

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$:

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in$$

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in \mathbb{R}.$$

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in \mathbb{R}.$$

We write φ in the expanded form:

$$\varphi(a) =$$

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in \mathbb{R}.$$

We write φ in the expanded form:

$$\varphi(a) = \mathbb{E}\Big(Y^2 \mid X = x\Big) - 2 \cdot a \cdot \mathbb{E}\Big(Y \mid X = x\Big) + a^2.$$

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in \mathbb{R}.$$

We write φ in the expanded form:

$$\varphi(a) = \mathbb{E}\Big(Y^2 \mid X = x\Big) - 2 \cdot a \cdot \mathbb{E}\Big(Y \mid X = x\Big) + a^2.$$

Now, we can calculate the min point of $\varphi(a)$ pretty easily, by solving $\varphi'(a) = 0$:

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in \mathbb{R}.$$

We write φ in the expanded form:

$$\varphi(a) = \mathbb{E}\Big(Y^2 \mid X = x\Big) - 2 \cdot a \cdot \mathbb{E}\Big(Y \mid X = x\Big) + a^2.$$

Now, we can calculate the min point of $\varphi(a)$ pretty easily, by solving $\varphi'(a) = 0$:

$$a = \mathbb{E}\Big(Y \mid X = x\Big) \qquad \text{is the only minimum point.}$$

Bayes Predictor in the LS Regression Problem

Let us fix x, the value of X, and find $g^*(x)$. This means that we need to minimize

$$\mathbb{E}\Big((Y - g(x))^2 \mid X = x\Big)$$

over $g(x)$, i.e., find the value $g(x)$, minimizing this Expectation. Let us use the notation $a = g(x)$: we want to minimize

$$\varphi(a) = \mathbb{E}\Big((Y - a)^2 \mid X = x\Big), \qquad a \in \mathbb{R}.$$

We write φ in the expanded form:

$$\varphi(a) = \mathbb{E}\Big(Y^2 \mid X = x\Big) - 2 \cdot a \cdot \mathbb{E}\Big(Y \mid X = x\Big) + a^2.$$

Now, we can calculate the min point of $\varphi(a)$ pretty easily, by solving $\varphi'(a) = 0$:

$$a = \mathbb{E}\Big(Y \mid X = x\Big) \qquad \text{is the only minimum point.}$$

So, finally, we have the Bayes Predictor in the LS Regression Problem:

$$g^*(x) = \mathbb{E}\Big(Y \mid X = x\Big), \qquad \forall x.$$

Bayes Predictor in the LS Regression Problem: Note

Note: Recall the minimization property of the Conditional Expectation: for any g,

$$\mathbb{E}\Big((Y - \mathbb{E}(Y|X))^2\Big) \leq \mathbb{E}\Big((Y - g(X))^2\Big).$$

Bayes Predictor in the LS Regression Problem: Note

Note: Recall the minimization property of the Conditional Expectation: for any g,

$$\mathbb{E}\Big((Y - \mathbb{E}(Y|X))^2\Big) \leq \mathbb{E}\Big((Y - g(X))^2\Big).$$

So we could use this: the Bayes Predictor for the Regression Problem is

Bayes Predictor in the LS Regression Problem: Note

Note: Recall the minimization property of the Conditional Expectation: for any g,

$$\mathbb{E}\Big((Y - \mathbb{E}(Y|X))^2\Big) \leq \mathbb{E}\Big((Y - g(X))^2\Big).$$

So we could use this: the Bayes Predictor for the Regression Problem is $g^*(X) = \mathbb{E}(Y|X)$, i.e.,

$$g^*(x) = \mathbb{E}(Y|X = x).$$

Bayes Predictor in the LS Regression Problem: Note

Note: Recall the minimization property of the Conditional Expectation: for any g,

$$\mathbb{E}\Big((Y - \mathbb{E}(Y|X))^2\Big) \leq \mathbb{E}\Big((Y - g(X))^2\Big).$$

So we could use this: the Bayes Predictor for the Regression Problem is $g^*(X) = \mathbb{E}(Y|X)$, i.e.,

$$g^*(x) = \mathbb{E}(Y|X = x).$$

Note: The function $g^*(x) = \mathbb{E}(Y|X = x)$ is called the Regression Function.

Bayes Predictor in the LS Regression Problem: Note

Note: Recall the minimization property of the Conditional Expectation: for any g,

$$\mathbb{E}\Big((Y - \mathbb{E}(Y|X))^2\Big) \leq \mathbb{E}\Big((Y - g(X))^2\Big).$$

So we could use this: the Bayes Predictor for the Regression Problem is $g^*(X) = \mathbb{E}(Y|X)$, i.e.,

$$g^*(x) = \mathbb{E}(Y|X = x).$$

Note: The function $g^*(x) = \mathbb{E}(Y|X = x)$ is called the Regression Function. It is solving the Minimization Problem:

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$$

Note: Geometric Interpretation:

Bayes Predictor in the L^1 Loss Regression Problem

Now, a slight modification of our Regression Problem: here we consider the following Loss function:

$$\ell(y_1, y_2) = |y_1 - y_2|.$$

Bayes Predictor in the L^1 Loss Regression Problem

Now, a slight modification of our Regression Problem: here we consider the following Loss function:

$$\ell(y_1, y_2) = |y_1 - y_2|.$$

Then, our Problem becomes

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$$

Bayes Predictor in the L^1 Loss Regression Problem

Now, a slight modification of our Regression Problem: here we consider the following Loss function:

$$\ell(y_1, y_2) = |y_1 - y_2|.$$

Then, our Problem becomes

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$$

Again we fix x, and think about finding $g^*(x)$, i.e. about solving the problem of minimizing

$$\varphi(a) =$$

Bayes Predictor in the L^1 Loss Regression Problem

Now, a slight modification of our Regression Problem: here we consider the following Loss function:

$$\ell(y_1, y_2) = |y_1 - y_2|.$$

Then, our Problem becomes

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$$

Again we fix x, and think about finding $g^*(x)$, i.e. about solving the problem of minimizing

$$\varphi(a) = \mathbb{E}\Big(\big|Y - a\big| \mid X = x\Big), \qquad a \in \mathbb{R}.$$

Bayes Predictor in the L^1 Loss Regression Problem

Now, a slight modification of our Regression Problem: here we consider the following Loss function:

$$\ell(y_1, y_2) = |y_1 - y_2|.$$

Then, our Problem becomes

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$$

Again we fix x, and think about finding $g^*(x)$, i.e. about solving the problem of minimizing

$$\varphi(a) = \mathbb{E}\Big(\big|Y - a\big| \mid X = x\Big), \qquad a \in \mathbb{R}.$$

It can be proven that the solution will be:

$$g^*(x) =$$

Bayes Predictor in the L^1 Loss Regression Problem

Now, a slight modification of our Regression Problem: here we consider the following Loss function:

$$\ell(y_1, y_2) = |y_1 - y_2|.$$

Then, our Problem becomes

$$g^* \in \underset{g}{argmin}\, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$$

Again we fix x, and think about finding $g^*(x)$, i.e. about solving the problem of minimizing

$$\varphi(a) = \mathbb{E}\Big(\big|Y - a\big| \mid X = x\Big), \qquad a \in \mathbb{R}.$$

It can be proven that the solution will be:

$$g^*(x) = Median(Y|X = x).$$

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$;
  - the solution was $g^*(x) = Median(Y|X = x)$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$;
  - the solution was $g^*(x) = Median(Y|X = x)$;
- For the $0 - 1$ Loss Classification Problem,
  - the Loss was $\ell(y_1, y_2) = \mathbf{1}(y_1 \neq y_2)$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$;
  - the solution was $g^*(x) = Median(Y|X = x)$;
- For the $0 - 1$ Loss Classification Problem,
  - the Loss was $\ell(y_1, y_2) = \mathbf{1}(y_1 \neq y_2)$;
  - the Problem was $g^* \in argmin_g \, \mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big) = argmin_g \, \mathbb{P}(Y \neq g(X))$;

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;
  - the Problem was $g^* \in argmin_g \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$;
  - the solution was $g^*(x) = Median(Y|X = x)$;
- For the $0 - 1$ Loss Classification Problem,
  - the Loss was $\ell(y_1, y_2) = \mathbf{1}(y_1 \neq y_2)$;
  - the Problem was $g^* \in argmin_g \mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big) = argmin_g \mathbb{P}(Y \neq g(X))$;
  - the solution was $g^*(x) =$

Summary

Let us summarize what we have obtained:

- For the L^2 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = (y_1 - y_2)^2$;
  - the Problem was $g^* \in argmin_g \mathbb{E}\Big((Y - g(\mathbf{X}))^2\Big)$;
  - the solution was $g^*(x) = \mathbb{E}(Y|X = x)$;
- For the L^1 Regression Problem,
  - the Loss was $\ell(y_1, y_2) = |y_1 - y_2|$;
  - the Problem was $g^* \in argmin_g \mathbb{E}\Big(\big|Y - g(\mathbf{X})\big|\Big)$;
  - the solution was $g^*(x) = Median(Y|X = x)$;
- For the $0 - 1$ Loss Classification Problem,
  - the Loss was $\ell(y_1, y_2) = \mathbf{1}(y_1 \neq y_2)$;
  - the Problem was $g^* \in argmin_g \mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big) = argmin_g \mathbb{P}(Y \neq g(X))$;
  - the solution was $g^*(x) = Mode(Y|X = x)$;

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$).

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$). Unfortunately, usually we do not know this Distribution, and what we have is just a realization of a Random Sample (Training Data) from that Distribution:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n).$$

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$). Unfortunately, usually we do not know this Distribution, and what we have is just a realization of a Random Sample (Training Data) from that Distribution:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n).$$

Now, what is the idea of having a good Prediction Algorithm?

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$). Unfortunately, usually we do not know this Distribution, and what we have is just a realization of a Random Sample (Training Data) from that Distribution:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n).$$

Now, what is the idea of having a good Prediction Algorithm? Good Algorithm will give, based on the Observations, a Predictor, the Risk of which is very close to the Ideal one: to the Bayes Risk.

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$). Unfortunately, usually we do not know this Distribution, and what we have is just a realization of a Random Sample (Training Data) from that Distribution:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n).$$

Now, what is the idea of having a good Prediction Algorithm? Good Algorithm will give, based on the Observations, a Predictor, the Risk of which is very close to the Ideal one: to the Bayes Risk. I.e., based on these Observations, we want to have a Predictor g_n (or, a Method to find that Predictor) such that

$$Risk(g_n) \approx Risk(g^*) = \text{Bayes Risk}.$$

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$). Unfortunately, usually we do not know this Distribution, and what we have is just a realization of a Random Sample (Training Data) from that Distribution:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n).$$

Now, what is the idea of having a good Prediction Algorithm? Good Algorithm will give, based on the Observations, a Predictor, the Risk of which is very close to the Ideal one: to the Bayes Risk. I.e., based on these Observations, we want to have a Predictor g_n (or, a Method to find that Predictor) such that

$$Risk(g_n) \approx Risk(g^*) = \text{Bayes Risk}.$$

Of course, we will always have

$$Risk(g_n) \geq Risk(g^*).$$

Back to Reality

So far we were constructing Bayes Predictors. But, usually we can find Bayes Predictors if we know the Joint Distribution of X and Y (or, at least, Conditional Distribution of $Y|X = x$). Unfortunately, usually we do not know this Distribution, and what we have is just a realization of a Random Sample (Training Data) from that Distribution:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n).$$

Now, what is the idea of having a good Prediction Algorithm? Good Algorithm will give, based on the Observations, a Predictor, the Risk of which is very close to the Ideal one: to the Bayes Risk. I.e., based on these Observations, we want to have a Predictor g_n (or, a Method to find that Predictor) such that

$$Risk(g_n) \approx Risk(g^*) = \text{Bayes Risk}.$$

Of course, we will always have

$$Risk(g_n) \geq Risk(g^*).$$

Now note that $Risk(g_n)$ is Radnom, since we construct g_n using

Good Algorithms

To define a good Learning Algorithms, let us assume that we have an infinite sequence of IID Observations from the Distribution behind X and Y:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), ...$$

Good Algorithms

To define a good Learning Algorithms, let us assume that we have an infinite sequence of IID Observations from the Distribution behind X and Y:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), ...$$

Assume we have an Algorithm $\mathcal{A}$, producing a Predictor g_n, when applied to the first n Observations.

Good Algorithms

To define a good Learning Algorithms, let us assume that we have an infinite sequence of IID Observations from the Distribution behind X and Y:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), ...$$

Assume we have an Algorithm $\mathcal{A}$, producing a Predictor g_n, when applied to the first n Observations. Then we say:

- $\mathcal{A}$ is **Consistent**, if

Good Algorithms

To define a good Learning Algorithms, let us assume that we have an infinite sequence of IID Observations from the Distribution behind X and Y:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), ...$$

Assume we have an Algorithm $\mathcal{A}$, producing a Predictor g_n, when applied to the first n Observations. Then we say:

- $\mathcal{A}$ is **Consistent**, if

$$Risk(g_n) \xrightarrow{\mathbb{P}} Risk(g^*), \qquad as \quad n \to +\infty.$$

Good Algorithms

To define a good Learning Algorithms, let us assume that we have an infinite sequence of IID Observations from the Distribution behind X and Y:

$$(X_1, Y_1), (X_2, Y_2), ..., (X_n, Y_n), ...$$

Assume we have an Algorithm $\mathcal{A}$, producing a Predictor g_n, when applied to the first n Observations. Then we say:

- $\mathcal{A}$ is **Consistent**, if

$$Risk(g_n) \xrightarrow{\mathbb{P}} Risk(g^*), \qquad as \quad n \to +\infty.$$

- $\mathcal{A}$ is **Universally Consistent**, if it is Consistent for all Probability Distributions over $\mathcal{X} \times \mathcal{Y}$, i.e., for all Possible Distributions of (X, Y).

Examples of Good Algorithms

We consider the Binary Classification Problem here.

Examples of Good Algorithms

We consider the Binary Classification Problem here.

Theorem (Stone): The k-NN Classifier is Universally Consistent,

if $k \to \infty$ and $n \to \infty$ in such a way that $\dfrac{k}{n} \to 0$.

Examples of Good Algorithms

We consider the Binary Classification Problem here.

Theorem (Stone): The k-NN Classifier is Universally Consistent, if $k \to \infty$ and $n \to \infty$ in such a way that $\dfrac{k}{n} \to 0$.

Theorem (Steinwart): Under some conditions, SVM is Universally Consistent.

Examples of Good Algorithms

We consider the Binary Classification Problem here.

Theorem (Stone): The k-NN Classifier is Universally Consistent, if $k \to \infty$ and $n \to \infty$ in such a way that $\dfrac{k}{n} \to 0$.

Theorem (Steinwart): Under some conditions, SVM is Universally Consistent.

We will talk about k-NN and SVM soon.

Methods to obtain some (good?) Algorithms, BC Problem

Consider again the 0-1 Binary Classification Algorithm.

Methods to obtain some (good?) Algorithms, BC Problem

Consider again the 0-1 Binary Classification Algorithm. The Problem was to minimize

$$\mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big) = \mathbb{P}(Y \neq g(X)).$$

Methods to obtain some (good?) Algorithms, BC Problem

Consider again the 0-1 Binary Classification Algorithm. The Problem was to minimize

$$\mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big) = \mathbb{P}(Y \neq g(X)).$$

Several Approaches to approximate this problem using the Dataset $(\mathbf{X}_k, Y_k)$, $k = 1, ..., n$.

a. Empirical Risk Minimization

We consider the Empirical Risk:

$ERM(g) =$

a. Empirical Risk Minimization

We consider the Empirical Risk:

$$ERM(g) = \frac{1}{n} \cdot \sum_{k=1}^{n} \mathbf{1}(Y_k \neq g(\mathbf{X_k})) =$$

a. Empirical Risk Minimization

We consider the Empirical Risk:

$$ERM(g) = \frac{1}{n} \cdot \sum_{k=1}^{n} \mathbf{1}(Y_k \neq g(\mathbf{X_k})) = \frac{\#\{Y_k \neq g(\mathbf{X_k}) : k = 1, ..., n\}}{n}.$$

a. Empirical Risk Minimization

We consider the Empirical Risk:

$$ERM(g) = \frac{1}{n} \cdot \sum_{k=1}^{n} \mathbf{1}(Y_k \neq g(\mathbf{X_k})) = \frac{\#\{Y_k \neq g(\mathbf{X_k}) : k = 1, ..., n\}}{n}.$$

By the LLN, we know that, for large n,

$$ERM(g) \approx \mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big).$$

a. Empirical Risk Minimization

We consider the Empirical Risk:

$$ERM(g) = \frac{1}{n} \cdot \sum_{k=1}^{n} \mathbf{1}(Y_k \neq g(\mathbf{X_k})) = \frac{\#\{Y_k \neq g(\mathbf{X_k}) : k = 1, ..., n\}}{n}.$$

By the LLN, we know that, for large n,

$$ERM(g) \approx \mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big).$$

Then the ERM strategy is, instead of Minimizing $\mathbb{E}\Big(\mathbf{1}(Y \neq g(\mathbf{X}))\Big)$, minimize the $ERM(g)$. Here we can fix some class of functions $\mathcal{G}$ and minimize $ERM(g)$ over $g \in \mathcal{G}$.

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) =$$

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) = \mathbf{1}\left(\eta(x) \geq \frac{1}{2}\right).$$

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) = \mathbf{1}\left(\eta(x) \geq \frac{1}{2}\right).$$

Here

$$\eta(x) =$$

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) = \mathbf{1}\left(\eta(x) \geq \frac{1}{2}\right).$$

Here

$$\eta(x) = \mathbb{E}(Y|X = x) =$$

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) = \mathbf{1}\left(\eta(x) \geq \frac{1}{2}\right).$$

Here

$$\eta(x) = \mathbb{E}(Y|X = x) = \mathbb{P}(Y = 1|X = x)$$

is the **Regression Function** (in the Classification setting).

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) = \mathbf{1}\left(\eta(x) \geq \frac{1}{2}\right).$$

Here

$$\eta(x) = \mathbb{E}(Y|X = x) = \mathbb{P}(Y = 1|X = x)$$

is the **Regression Function** (in the Classification setting).

Now, the Regression Function Approximation strategy is to approximate the Regression Function $\eta(x)$ by some Estimate $\hat{\eta}(x)$, and take

$$g(x) =$$

b. Regression Function Approximation

We know the solution of the BinClass Classification Problem, the Bayes Predictor:

$$g^*(x) = \mathbf{1}\left(\eta(x) \geq \frac{1}{2}\right).$$

Here

$$\eta(x) = \mathbb{E}(Y|X = x) = \mathbb{P}(Y = 1|X = x)$$

is the **Regression Function** (in the Classification setting).

Now, the Regression Function Approximation strategy is to approximate the Regression Function $\eta(x)$ by some Estimate $\hat{\eta}(x)$, and take

$$g(x) = \mathbf{1}\left(\hat{\eta}(x) \geq \frac{1}{2}\right).$$

Regression Function Approximation, cont'd

Some possible (standard) ways to take $\hat{\eta}(x)$:

Regression Function Approximation, cont'd

Some possible (standard) ways to take $\hat{\eta}(x)$:

- $\eta(x) = \mathbb{P}(Y = 1 | X = x) \approx$

Regression Function Approximation, cont'd

Some possible (standard) ways to take $\hat{\eta}(x)$:

- $\eta(x) = \mathbb{P}(Y = 1 | X = x) \approx \dfrac{\#\{Y_k = 1 \ : \ X_k = x\}}{\#\{X_k = x\}} = \hat{\eta}(x)$

Regression Function Approximation, cont'd

Some possible (standard) ways to take $\hat{\eta}(x)$:

- $\eta(x) = \mathbb{P}(Y = 1 | X = x) \approx \dfrac{\#\{Y_k = 1 \ : \ X_k = x\}}{\#\{X_k = x\}} = \hat{\eta}(x)$

The Algorithm: In this case the Algorithm will be: OTB

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that

$$\eta(x) = \mathbb{E}(Y|X = x),$$

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that
$$\eta(x) = \mathbb{E}(Y|X = x),$$
so $\eta(x)$ is the Average of all Y-s with $X = x$.

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that

$$\eta(x) = \mathbb{E}(Y|X = x),$$

  so $\eta(x)$ is the Average of all Y-s with $X = x$. In practice, when we have a Dataset (X_k, Y_k), we can have 0 or small number of points with $X_k = x$ (say, if our new observation X will have different value than all X_k-s), then the previous approach will not work.

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that
$$\eta(x) = \mathbb{E}(Y|X = x),$$
so $\eta(x)$ is the Average of all Y-s with $X = x$. In practice, when we have a Dataset (X_k, Y_k), we can have 0 or small number of points with $X_k = x$ (say, if our new observation X will have different value than all X_k-s), then the previous approach will not work. So we take some $k \in \mathbb{N}$, denote
$$NN_k(x) = \text{The set of } k \text{ Nearest Points } X_k \text{ from } x,$$
and we take
$$\hat{\eta}(x) = Average\Big(Y_i \mid X_i \in NN_k(x)\Big)$$

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that
$$\eta(x) = \mathbb{E}(Y|X = x),$$
so $\eta(x)$ is the Average of all Y-s with $X = x$. In practice, when we have a Dataset (X_k, Y_k), we can have 0 or small number of points with $X_k = x$ (say, if our new observation X will have different value than all X_k-s), then the previous approach will not work. So we take some $k \in \mathbb{N}$, denote
$$NN_k(x) = \text{The set of } k \text{ Nearest Points } X_k \text{ from } x,$$
and we take
$$\hat{\eta}(x) = Average\Big(Y_i \mid X_i \in NN_k(x)\Big)$$
This is the idea of the k-NN.

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that
$$\eta(x) = \mathbb{E}(Y|X = x),$$
so $\eta(x)$ is the Average of all Y-s with $X = x$. In practice, when we have a Dataset (X_k, Y_k), we can have 0 or small number of points with $X_k = x$ (say, if our new observation X will have different value than all X_k-s), then the previous approach will not work. So we take some $k \in \mathbb{N}$, denote
$$NN_k(x) = \text{The set of } k \text{ Nearest Points } X_k \text{ from } x,$$
and we take
$$\hat{\eta}(x) = Average\Big(Y_i \mid X_i \in NN_k(x)\Big)$$
This is the idea of the k-NN. Of course, we need to define a distance to calculate the Nearest Points to x.

Regression Function Approximation, cont'd

Another standard way to appoximate $\eta(x)$:

- Recall that
$$\eta(x) = \mathbb{E}(Y|X = x),$$
so $\eta(x)$ is the Average of all Y-s with $X = x$. In practice, when we have a Dataset (X_k, Y_k), we can have 0 or small number of points with $X_k = x$ (say, if our new observation X will have different value than all X_k-s), then the previous approach will not work. So we take some $k \in \mathbb{N}$, denote
$$NN_k(x) = \text{The set of } k \text{ Nearest Points } X_k \text{ from } x,$$
and we take
$$\hat{\eta}(x) = Average\Big(Y_i \mid X_i \in NN_k(x)\Big)$$
This is the idea of the k-NN. Of course, we need to define a distance to calculate the Nearest Points to x.

The Algorithm: In this case the Algorithm will be: OTB

c. Density Estimation

Recall the other representation for the Binary Classification Bayes Classifier, obtained using the Bayes Rule:

c. Density Estimation

Recall the other representation for the Binary Classification Bayes Classifier, obtained using the Bayes Rule: we take $g^*(x) = 1$ iff

$$\mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0).$$

c. Density Estimation

Recall the other representation for the Binary Classification Bayes Classifier, obtained using the Bayes Rule: we take $g^*(x) = 1$ iff

$$\mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0).$$

Now, in the Density Estimation approach,

- we Estimate $\mathbb{P}(Y = 1)$ by

c. Density Estimation

Recall the other representation for the Binary Classification Bayes Classifier, obtained using the Bayes Rule: we take $g^*(x) = 1$ iff

$$\mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0).$$

Now, in the Density Estimation approach,

- we Estimate $\mathbb{P}(Y = 1)$ by $\dfrac{\sum_{k=1}^{n} Y_k}{n}$,

c. Density Estimation

Recall the other representation for the Binary Classification Bayes Classifier, obtained using the Bayes Rule: we take $g^*(x) = 1$ iff

$$\mathbb{P}(X = x | Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x | Y = 0) \cdot \mathbb{P}(Y = 0).$$

Now, in the Density Estimation approach,

- we Estimate $\mathbb{P}(Y = 1)$ by $\frac{\sum_{k=1}^{n} Y_k}{n}$, and, of course, then we will take $\left(1 - \frac{\sum_{k=1}^{n} Y_k}{n}\right)$ for $\mathbb{P}(Y = 0)$;

c. Density Estimation

Recall the other representation for the Binary Classification Bayes Classifier, obtained using the Bayes Rule: we take $g^*(x) = 1$ iff

$$\mathbb{P}(X = x|Y = 1) \cdot \mathbb{P}(Y = 1) \geq \mathbb{P}(X = x|Y = 0) \cdot \mathbb{P}(Y = 0).$$

Now, in the Density Estimation approach,

- we Estimate $\mathbb{P}(Y = 1)$ by $\dfrac{\sum_{k=1}^{n} Y_k}{n}$, and, of course, then we will take $\left(1 - \dfrac{\sum_{k=1}^{n} Y_k}{n}\right)$ for $\mathbb{P}(Y = 0)$;
- we assume some (say, Parametric) Model behind the Distributions

$$X|Y = 1 \qquad \textit{and} \qquad X|Y = 0,$$

Estimate these Distributions, and calculate/approximate $\mathbb{P}(X = x|Y = 1)$ and $\mathbb{P}(X = x|Y = 0)$ by densities of $X|Y = 1$ and $X|Y = 0$, respectively.

c. Density Estimation, cont'd

So, if $f_1(x)$ and $f_0(x)$ are Estimated Densities for

$$X|Y = 1 \qquad \textit{and} \qquad X|Y = 0,$$

c. Density Estimation, cont'd

So, if $f_1(x)$ and $f_0(x)$ are Estimated Densities for

$$X|Y = 1 \qquad and \qquad X|Y = 0,$$

then we will take $g(x) = 1$ iff

$$f_1(x) \cdot \frac{\sum_{k=1}^{n} Y_k}{n} \geq f_0(x) \cdot \left(1 - \frac{\sum_{k=1}^{n} Y_k}{n}\right).$$

c. Density Estimation, cont'd

So, if $f_1(x)$ and $f_0(x)$ are Estimated Densities for

$$X|Y = 1 \qquad and \qquad X|Y = 0,$$

then we will take $g(x) = 1$ iff

$$f_1(x) \cdot \frac{\sum_{k=1}^{n} Y_k}{n} \geq f_0(x) \cdot \left(1 - \frac{\sum_{k=1}^{n} Y_k}{n}\right).$$

This approach is giving the LDA, QDA.