Machine Learning

Lecture 9 - Mutual Information Neural Estimation

Lecturer: Haim Permuter *Scribe: Yonatan Dadon, Cameron Solomon*

I. Introduction

In this lecture we introduce an estimation method for the Mutual Information between two random variables using the power of neural networks. First, we recall the required definitions from information theory, and expand on their properties. Then, we introduce a new and a very useful way of representing information measures, which is called the variational formulation. Using the variational formulation we will be able to apply maximization methods that we have previously applied in Machine Learning algorithms and, hence, to develop an iterative algorithm that will output an estimation of Mutual Information. Lastly, we will consolidate our understanding of this methodology and view it from the the standpoint of Hypothesis Testing.

II. Divergence and Mutual information

Definition 1 (Kullback Liebler Divergence) The *Kullback Liebler Divergence* between two probability densities $P(x), Q(x)$ is defined as:

$$\begin{aligned} D_{KL}(P||Q) &\triangleq \mathbb{E}_P\left[\log\frac{P(X)}{Q(X)}\right] \\ &= \int_{\mathcal{X}} P(x)\log\frac{P(x)}{Q(x)}dx. \end{aligned} \tag{1}$$

Remark 1 (Non-negativity of the Kullback Liebler Divergence) For any two distributions P, Q the Kullback Liebler Divergence is non-negative. i.e.

$$D_{KL}(P||Q) \geq 0. \tag{2}$$

And equality holds if and only if $P(x) = Q(x), \quad \forall x \in \mathcal{X}$.

Definition 2 (Mutual Information) Let X and Y be two random variables with a joint distribution $P(x, y)$. The *Mutual Information* $I(X;Y)$ is defined as

$$I(X;Y) \triangleq \mathbb{E}_{P_{XY}} \left[\log \frac{P(X,Y)}{P(X)P(Y)} \right]. \tag{3}$$

Remark 2 (Defining Mutual Information using Kullback Liebler Divergence) Mutual Information can be easily defined using the Kullback Liebler Divergence as follows:

$$I(X;Y) = D_{KL}(P_{XY}||P_X P_Y). \tag{4}$$

Remark 3 From the previous remark it is easy to show that $I(X;Y) = 0$ if and only if X, Y are statistically independent.

III. The Donsker-Varadhan variational formula

In this section we introduce a new and a very useful way of representing the Kullback Liebler Divergence ,which is called the *variational formulation*. The variation formulation is a way of representing some measures as a supremum or infimum over a set of functions. A general form of variational formulation is as follows:

$$f(x) = \sup_{\lambda} F_\lambda(x). \tag{5}$$

This representation has some distinct advantages. First, it provides upper / lower bounds for the represented measure which could not be obtained beforehand. Second, in many cases the variational formulation might be easier to compute analytically. Third, the use of optimization methods in order to achieve certain approximations might be a handy solution. The following theorem introduces a variational formulation for the Kullback Liebler Divergence, and perform as a key ingredient of neural estimation of Mutual Information.

Theorem 1 (Donsker-Varadhan representation [1])
Let X be a random variable with alphabet $\mathcal{X}$ and let P, Q be two probability density

functions. The Kullback Liebler Divergence admits the following dual representation:

$$D_{KL}(P||Q) = \sup_{T:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[T(X)] - \log(\mathbb{E}_Q[e^{T(X)}]). \tag{6}$$

Proof:

The proof consists of two parts which we will formulate and prove in the following two lemmas:

Lemma 1 (Existence of supremum in Donsker-Varadhan variational representation) There exists a function $T^* : \mathcal{X} \to \mathbb{R}$ such that:

$$D_{KL}(P||Q) = \mathbb{E}_P[T^*(X)] - \log(\mathbb{E}_Q[e^{T^*(X)}]). \tag{7}$$

Proof:

Let us choose $T^*(x) = \log \frac{P(x)}{Q(x)}$. Note that the following series of equalities hold:

$$\begin{aligned}
\mathbb{E}_P[T^*(X)] - \log(\mathbb{E}_Q[e^{T^*(X)}]) &\overset{(a)}{=} \mathbb{E}_P\left[\log \frac{P(X)}{Q(X)}\right] - \log\left(\mathbb{E}_Q\left[e^{\log \frac{P(X)}{Q(X)}}\right]\right) && (8)\\
&\overset{(b)}{=} D_{KL}(P||Q) - \log\left(\mathbb{E}_Q\left[\frac{P(X)}{Q(X)}\right]\right) && (9)\\
&\overset{(c)}{=} D_{KL}(P||Q) - \log(\int_{\mathcal{X}} Q(x)\frac{P(x)}{Q(x)}dx) && (10)\\
&= D_{KL}(P||Q) - \log(\int_{\mathcal{X}} P(x)dx) && (11)\\
&\overset{(d)}{=} D_{KL}(P||Q) - \log(1) && (12)\\
&= D_{KL}(P||Q). && (13)
\end{aligned}$$

where

(a) Follows from the specific choice of $T^*(x) = \log \frac{P(x)}{Q(x)}$.

(b) Follows from the definition of the Kullback Liebler Divergence.

(c) Follows from the definition of the expectation of continuous random variable.

(d) Integration of any probability density is always 1.

■

Lemma 2 (Lower bound for the Kullback Liebler Divergence) For any function $T : \mathcal{X} \to \mathbb{R}$ the following inequality holds:

$$D_{KL}(P||Q) \geq \mathbb{E}_P[T(X)] - \log(\mathbb{E}_Q[e^{T(X)}]). \tag{14}$$

Proof:

Let us define a new probability density function by:

$$G(x) \triangleq \frac{Q(x)e^{T(x)}}{\mathbb{E}_Q[e^{T(X)}]}. \tag{15}$$

Note that $G(x) \geq 0$ and forms a probability density function since

$$\int_{\mathcal{X}} G(x)dx = \int_{\mathcal{X}} \frac{Q(x)e^{T(x)}}{\mathbb{E}_Q[e^{T(X)}]}dx = \frac{\mathbb{E}_Q[e^{T(X)}]}{\mathbb{E}_Q[e^{T(X)}]} = 1. \tag{16}$$

Back to the proof, we use $G(x)$ to obtain:

$$\begin{aligned}
D_{KL}(P||Q) - \mathbb{E}_P\left[T(X)\right] + \log(\mathbb{E}_Q\left[e^{T(X)}\right]) &\overset{(a)}{=} \mathbb{E}_P\left[\log\frac{P(X)}{Q(X)} - T(X)\right] + \log(\mathbb{E}_Q\left[e^{T(X)}\right]) \\
&\overset{(b)}{=} \mathbb{E}_P\left[\log\frac{P(X)}{Q(X)e^{T(X)}} + \log(\mathbb{E}_Q[e^{T(X)}])\right] \\
&= \mathbb{E}_P\left[\log\frac{P(X)\mathbb{E}_Q[e^{T(X)}]}{Q(X)e^{T(X)}}\right] \\
&\overset{(c)}{=} \mathbb{E}_P\left[\log\frac{P(X)}{G(X)}\right] \\
&= D_{KL}(P||G) \\
&\overset{(d)}{\geq} 0.
\end{aligned}$$

where

(a) Follows from the definition of Divergence and linearity of expectation.

(b) Follows from the fact that $\log(\mathbb{E}_Q[e^{T(X)}])$ is a deterministic constant.

(c) Follows from the definition of $G(x) = \frac{Q(x)e^{T(x)}}{\mathbb{E}_Q[e^{T(X)}]}$.

(d) Follows from the non-negativity of Kullback Liebler Divergence (see Definition 1 in section II).

■

Now, back to the proof of Theorem 1 (Donsker-Varadhan representation):
We showed that by choosing $T^*(x) = \log \frac{P(x)}{Q(x)}$ we obtain:

$$D_{KL}(P||Q) = \mathbb{E}_P\left[T^*(X)\right] - \log(\mathbb{E}_Q[e^{T^*(X)}]). \tag{17}$$

We also proved that for any function $T : \mathcal{X} \to \mathbb{R}$ the following holds:

$$D_{KL}(P||Q) \geq \mathbb{E}_P[T(X)] - \log(\mathbb{E}_Q[e^{T(X)}]). \tag{18}$$

Hence,

$$D_{KL}(P||Q) = \sup_{T:\mathcal{X}\to\mathbb{R}} \mathbb{E}_P[T(X)] - \log(\mathbb{E}_Q[e^{T(X)}]). \tag{19}$$

■

IV. Mutual Information Neural Estimation Algorithm (MINE)

In this section we will use the Donsker-Varadhan variational formulation in order to estimate the Mutual Information using neural networks. Using the results from Theorem 1 in the previous section we can write the Mutual Information in its variational form as:

$$I(X;Y) = \sup_{T:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \mathbb{E}_{P_{XY}}[T(X,Y)] - \log(\mathbb{E}_{P_X P_Y}[e^{T(X,Y)}]). \tag{20}$$

Problem definition:
Let $X \sim P_X$ and $Y \sim P_Y$ be two random variables with alphabets $\mathcal{X}, \mathcal{Y}$, respectively, and let $(X_i, Y_i)_{i=1}^n \sim P_{XY}$ be i.i.d samples. Our goal is to estimate the Mutual Information $I(X;Y)$ from the n given samples.

First, we may notice the following difficulties:
1) Note that in order to evaluate the Mutual Information in its variational form one requires a full knowledge of the joint and marginal distributions of X and Y. In practice these distributions may well be unknown.
2) Note that a maximization over all possible functions $T : \mathcal{X} \times \mathcal{Y} \to \mathbb{R}$ is required, which may be impractical in reality.

In order to overcome the first challenge we assumed a set of i.i.d samples which are drawn according to P_{XY} is given. Under the assumption that the number of samples n is large enough, we can use the Law of Large Numbers to obtain the following approximation:

$$\mathbb{E}_{P_{XY}}[T(X,Y)] \approx \frac{1}{n}\sum_{i=1}^{n} T(X_i, Y_i). \tag{21}$$

Note that an evaluation of $\log(\mathbb{E}_{P_X P_Y}[e^{T(X,Y)}])$ is still required. Now we are facing a new challenge since the given samples (X_i, Y_i) are drawn according to P_{XY} and not according to $P_X P_Y$. Therefore, direct use of the Law of Large Number is not correct. This can still be overcome by artificially constructing tuples of the form $(X_i, \widetilde{Y}_i)$, where $\widetilde{Y}_i$ is taken from the **randomly shuffled** or $\mathbf{randomlypermute}$ the set of all samples $(Y_i)_{i=1}^n$. Because (X_i, Y_i) is i.i.d., randomly shuffling Y_i will generate $\widetilde{Y}_i$ that is statistically independent of X_i and with the same pdf as Y_i, i.e., P_Y. Hence, we obtain that $(X_i, \widetilde{Y}_i)_{i=1}^n$ are i.i.d samples distributed according to $P_X P_Y$. Now it is possible to use the Law of Large Numbers to obtain the following approximation:

$$\log\left(\mathbb{E}_{P_X P_Y}[e^{T(X,Y)}]\right) \approx \log\left[\frac{1}{n}\sum_{i=1}^{n} e^{T(X_i, \widetilde{Y}_i)}\right]. \tag{22}$$

Finally, using equations (21) and (22) we shall define our Mutual Information estimator as:

$$\hat{I}(X;Y) \triangleq \sup_{T:\mathcal{X}\times\mathcal{Y}\to\mathbb{R}} \frac{1}{n}\sum_{i=1}^{n} T(X_i, Y_i) - \log\left[\frac{1}{n}\sum_{i=1}^{n} e^{T(X_i, \widetilde{Y}_i)}\right]. \tag{23}$$

This problem now reduces to finding the function $T(X,Y)$ that maximizes equation (23). To solve this maximization problem, we construct a neural network with parameters θ, which gets as its input the samples $(X_i,Y_i)_{i=1}^n$ and $(X_i,\widetilde{Y}_i)_{i=1}^n$; we can regard the function $T_\theta(X,Y)$ as the output of the neural network The network's cost function is defined as follows:

$$\hat{I}_\theta(X;Y)=\frac{1}{n}\sum_{i=1}^{n}T_\theta(X_i,Y_i)-\log\left(\frac{1}{n}\sum_{i=1}^{n}e^{T_\theta(X_i,\widetilde{Y}_i)}\right). \tag{24}$$

It is our goal to maximize this function. The back propagation algorithm makes it possible for us to compute partial derivatives with respect to the network's parameters θ, and then use the gradient ascent algorithm in order to move step by step towards local maxima of $\hat{I}_\theta(X;Y)$. Since, practically speaking, the number of given samples is large, we may use mini-batch gradient ascent in order to train the network to reach a local maximum of $\hat{I}_\theta(X;Y)$ by adjusting the network's parameters θ. The algorithm steps are as follows:

Algorithm 1 Mutual Information Neural Estimation Algorithm (MINE) [1]

1: $\theta \leftarrow$ Network parameters initialization.
2: **repeat**
3: Draw mini-batch of samples: $(X_1,Y_1),(X_2,Y_2),...,(X_m,Y_m)\sim P_{XY}$.
4: Randomly shuffle $Y_1,Y_2,...,Y_m$ to generate $\widetilde{Y_1},\widetilde{Y_2},...,\widetilde{Y_m}$
5: Draw m samples from the marginal distribution: $\widetilde{Y_1},\widetilde{Y_2},...,\widetilde{Y_m}\sim P_Y$.
6: Evaluate: $\hat{I}_\theta(X;Y)\leftarrow\frac{1}{m}\sum_{i=1}^{m}T_\theta(X_i,Y_i)-\log(\frac{1}{m}\sum_{i=1}^{m}e^{T_\theta(X_i,\widetilde{Y}_i)})$.
7: Update network parameters: $\theta\leftarrow\theta+\nabla_\theta\hat{I}_\theta(X;Y)$.
8: **until** convergence

V. Hypothesis Testing

A. Introduction

In the previous sections, we developed a way to approximate the Kullback Liebler Divergence using the Donsker-Varadhan representation. While doing this, we found that the the optimal function which maximizes the Donsker-Varadhan representation is given by $\log \frac{P(X)}{Q(X)}$, where $P(X)$ and $Q(X)$ are two probability functions from which our samples are generated. We then concluded that by optimizing over all neural Networks, we could obtain a network that would give us this logarithm as its output.

In this section, our goal is to identify from which distribution the samples $(x_1, x_2, ..., x_n)$ that we feed into our network were generated. Later on in this section, we will see that in order to do this, we will have to check the condition $\frac{P(X^n)}{Q(X^n)} > T$ for some threshold T. Since the samples are i.i.d, we can re-write:

$$P(x^n) = \prod_{i=1}^{n} P(x_i)$$
$$Q(x^n) = \prod_{i=1}^{n} Q(x_i),$$

and since logarithms are monotonically increasing functions, we can take the logarithm of both sides of the inequality written above and receive:

$$\begin{aligned}\log\left(\frac{P(x^n)}{Q(x^n)}\right) &\overset{(a)}{=} \log\left(\prod_{i=1}^{n}\frac{P(x_i)}{Q(x_i)}\right)\\ &\overset{(b)}{=} \sum_{i=1}^{n}\log\left(\frac{P(x_i)}{Q(x_i)}\right)\\ &> \log(T),\end{aligned}$$

where (a) follows from the fact that the samples are i.i.d and (b) follows from the fact that the logarithm of a product is the sum of the individual logarithms.

But $\log\left(\frac{P(x_i)}{Q(x_i)}\right)$ is the output of the neural network that we found through the optimization problem presented in the previous sections. Hence, when using the neural network to approximate the Kullback Liebler Divergence, we can simultaneously sum over the network's outputs, as a byproduct, find from which distribution our samples were generated!

B. Hypothesis Testing

We will now discuss how we can identify from which distribution our samples were generated and why the inequality given in the introduction provides us with the optimal decision region.

A set of samples $(x_1, x_2, ..., x_n)$ are given and our goal is to decide from which of two given distributions, $P_1(x)$ or $P_2(x)$, the samples were generated. In order to do this, we first define two hypotheses:

$$H_1 : X \sim P_1,$$

$$H_2 : X \sim P_2.$$

In addition, we define two forms of errors:

$$\alpha = P(H_2|X \text{ is from } P_1),$$

$$\beta = P(H_1|X \text{ is from } P_2).$$

α is the resultant error when we decided that X was generated from $P_2(X)$ when it was actually generated from $P_1(X)$, and β is the resultant error when we decided that X was generated from $P_1(X)$ when it was actually generated from $P_2(X)$.

We will now also define a function of the samples, $G : x^n \rightarrow 1, 2$, to be equal to 1 when our hypothesis for the samples is H_1, and to be equal to 2 when when our hypothesis for the samples is H_2. We can now re-write α and β with the help of $G(x^n)$:

$$\alpha = P(G(x^n) = 2 \ |H_1 \text{ is true}),$$

$$\beta = P(G(x^n) = 1 \ |H_2 \text{ is true}).$$

Remark 4 From this representation of α and β, we can see that there is a trade-off. As α decreases in value, β increases and vice-versa. For example, if α were to be small, $G(x^n)$ would have to be equal to 1 most of the time, thus causing β to increase.

Now that we have defined our errors, we must determine the decision region in order to

decide from which distribution our samples were generated.

Let us define the decision region:

$$\mathcal{A}_n(T) \triangleq \left\{ x^n : \frac{P_1(x^n)}{P_2(x^n)} > T \right\}.$$

Using the decision region given above, our decision criteria will be that if $X \in \mathcal{A}_n$, meaning that $\frac{P_1(x^n)}{P_2(x^n)} > T$, then X is generated by $P_1(X)$, and if $X \notin \mathcal{A}_n$, meaning that $\frac{P_1(x^n)}{P_2(x^n)} < T$, then X is generated by $P_2(X)$.

The following lemma will guarantee that the decision region defined above is the optimal region with regards to minimum error.

Lemma 3 (Neyman-Pearson Lemma [2])

We define:

$$\alpha^* = P_1(\mathcal{A}_n^c(T)) = P(\text{decide } H_2 | H_1 \text{ is true}),$$

$$\beta^* = P_2(\mathcal{A}_n(T)) = P(\text{decide } H_1 | H_2 \text{ is true}),$$

to be the errors of the optimal decision region, $\mathcal{A}_n$.

Let $\mathcal{B}_n$ be any other decision region with errors α and β.
Then, if $\alpha < \alpha^*$, β^* must be smaller than β.

Proof:

Let us first define two indicator function, $\phi_A(X)$ and $\phi_B(X)$, which will get the value 0 or 1 according to which of the two decision regions X belongs to. The explicit definition of these functions is given by:

$$\phi_A(X) = \begin{cases} 1, & \text{X} \in \mathcal{A} \\ 0, & \text{X} \notin \mathcal{A} \end{cases}, \quad \phi_B(X) = \begin{cases} 1, & \text{X} \in \mathcal{B} \\ 0, & \text{X} \notin \mathcal{B} \end{cases}$$

Let us consider

$$(\phi_A(x) - \phi_B(x)) * (P_1(x) - T * P_2(x)) \geq 0.$$

If $x \in \mathcal{A}$, $P_1(x) > T * P_2(x)$ and $(\phi_A(x) - \phi_B(x)) \geq 0$. If $x \notin \mathcal{A}$, $P_1(x) < T * P_2(x)$ and $(\phi_A(x) - \phi_B(x)) \leq 0$. In both cases, the product is greater than or equal to zero. Let

us now take the sum of these products over all values of x:

$$\begin{aligned}
&\sum_x (\phi_A(x) - \phi_B(x)) * (P_1(x) - T * P_2(x)) \\
&\qquad \stackrel{(a)}{=} \sum_x \phi_A(x) * (P_1(x) - T * P_2(x)) - \phi_B(x) * (P_1(x) - T * P_2(x)) \\
&\qquad \stackrel{(b)}{=} \sum_{x \in \mathcal{A}} (P_1(x) - T * P_2(x)) - \sum_{x \in \mathcal{B}} (P_1(x) - T * P_2(x)) \\
&\qquad \stackrel{(c)}{=} \sum_{x \in \mathcal{A}} P_1(x) - \sum_{x \in \mathcal{A}} T * P_2(x) - \sum_{x \in \mathcal{B}} P_1(x) + \sum_{x \in \mathcal{B}} T * P_2(x) \\
&\qquad \stackrel{(d)}{=} 1 - \alpha^* - T * \beta^* - 1 + \alpha + T * \beta \\
&\qquad \stackrel{(e)}{=} (\alpha - \alpha^*) + T * (\beta - \beta^*) \\
&\qquad \stackrel{(f)}{\geq} 0,
\end{aligned}$$

where

(a) Is obtained by opening the parentheses of the product.

(b) Is obtained by splitting the sum into two by summing over $\mathcal{A}$ and $\mathcal{B}$ and by remembering that $\phi_A(x)$ and $\phi_B(x)$ are indicators for each of these two groups.

(c) Follows from the linearity of the sums.

(d) Is obtained as follows:

Let us first consider

$$\textstyle\sum_{x \in \mathcal{A}} P_1(x).$$

Summing over the probability $P_1(X)$ when $X \in \mathcal{A}$ is the same as taking one minus the sum of the probability $P_1(X)$ when $X \notin \mathcal{A}$. But this sum is the exact definition of α^* that was given earlier since we are taking the probability of X with regards to $P_1(X)$ when it was really generated by $P_2(X)$. Therefore, we can write the sum as follows:

$$\textstyle\sum_{x \in \mathcal{A}} P_1(x) = 1 - \sum_{x \notin \mathcal{A}} P_1(x) = 1 - \alpha^*.$$

Now let us consider

$$\textstyle\sum_{x \in \mathcal{A}} T * P_2(x).$$

This time, we are summing over the probability $P_2(X)$ when X was really generated from $P_1(X)$. But this is the exact definition of β^* which was given previously. Therefore,

$$\sum_{x\in\mathcal{A}} T * P_2(x) = T * \beta^*.$$

From the explanations given above, we can now write

$$\sum_{x\in\mathcal{A}}(P_1(x) - T * P_2(x)) = 1 - \alpha^* - T * \beta^*.$$

As for the sum over $\mathcal{B}$, in the exact manner explained above, we can show that

$$\sum_{x\in\mathcal{B}}(P_1(x) - T * P_2(x)) = 1 - \alpha - T * \beta.$$

Our equality is now obtained by substituting the sums for the equations derived above.
(e) Is obtained by tidying up the equation.
(f) Follows from the fact that the equation that we were originally summing,

$$(\phi_A(x) - \phi_B(x)) * (P_1(x) - T * P_2(x)),$$

is non-negative. So therefore the sum is also non-negative.

We now have the following inequality:

$$(\alpha - \alpha^*) + T * (\beta - \beta^*) \geq 0.$$

If we look at this phrase, it is easy to see that if $\alpha < \alpha^*$, then β must be greater than β^* in order for the inequality to hold. This concludes our proof. ■

Remark 5 As a result of this lemma, we can conclude that the decision region defined by $\mathcal{A}_n$ is the optimal decision region for deciding from which probability distribution our samples were generated. Let us plot an example of the error defined by the optimal decision region $\mathcal{A}_n$ on a two-dimensional grid with axes α and β representing our two forms of error.

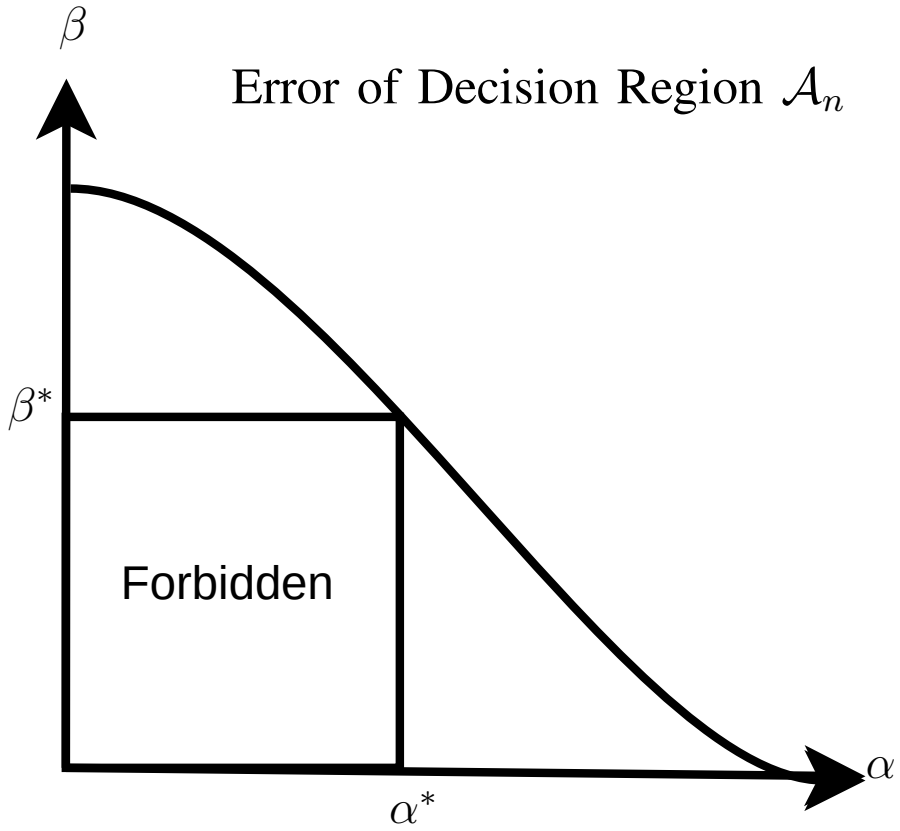

Fig. 1. The error given by $\mathcal{A}_n$ with an illegal region drawn beneath it.

If we were to pick any point on the plot of the error given by $\mathcal{A}_n$ and draw two perpendicular lines to each of the axes, as is shown in the plot, then any other decision region's error would have to fall outside this region, which can be seen in the plot as the area enclosed by the square. This is due to the fact that the Neyman-Pearson lemma tells us that if the error in one axis is smaller than the error of the optimal region in that same axis (e.g. $\alpha < \alpha^*$), then the error in the other axis must be larger (in this case, $\beta > \beta^*$). Since the point on the line is chosen arbitrarily, we can conclude that at no point beneath the plot of the error defined by our optimal decision region can there be an error from another decision region. Therefore, if we were to plot the error given by any other decision region, the plot must be above the plot of the error defined by $\mathcal{A}_n$, thus showing us that this is indeed the optimal decision region.

REFERENCES

[1] M. Belghazi, A. Baratin, S. Rajeswar, S. Ozair, Y. Bengio, A. Courville, and R.D Hjelm. Mine: mutual information neural estimation. *arXiv preprint arXiv:1801.04062*, 2018.

[2] T. M. Cover and J. A. Thomas. *Elements of information theory*, chapter 11, pages 375–379. John Wiley & Sons, second edition, 2012.