CS103 Winter 2019 – Assignment 2

Due: Monday February 25, 2019

Notation. Recall that we defined the following information theoretic quantities:

1. **Entropy:** $H_p(x) = \mathbb{E}_{x \sim p(x)}[-\log p(x)]$ (the suffix p is omitted if there is no ambiguity)
2. **Conditional-entropy:** $H_p(y|x) = \mathbb{E}_{x \sim p(x)} \mathbb{E}_{y \sim p(y|x)}[-\log p(y|x)]$
3. **KL-divergence:** $\mathrm{KL}(p(x) \,\|\, q(x)) = \mathbb{E}_{x \sim p(x)}[\log \frac{p(x)}{q(x)}]$ (recall that the KL divergence is ≥ 0, and equal to zero if and only if the distributions are the same).
4. **Mutual information:** $I(x, y) = H(x) - H(x|y)$
5. **Conditional mutual information:** $I(x, y|z) = H(x|z) - H(x|y, z)$

Exercise 1 (Information theoretic identities and properties, 10 points). (a) Using the above definitions, prove the following information theoretic identities:

1. $H(y|x) = H(x, y) - H(x)$;
2. $I(x; y) = H(x) + H(y) - H(x, y)$;
3. $I(x; y) = \mathbb{E}_{x \sim p(x)}[\mathrm{KL}(p(y|x) \,\|\, p(y))]$;

(b) Using the above identities prove that:

1. Mutual information is symmetric, that is $I(x; y) = I(y; x)$;
2. Mutual information is always positive;
3. Conditioning on a random variable does not increase: $H(y|x) \leq H(y)$ for any x and y;
4. Two variables x and y are independent ($p(x, y) = p(x)p(y)$ or equivalently $p(y|x) = p(y)$) if and only if $I(x; y) = 0$.

Exercise 2 (Learning a separating factor, 10 points). Often two inputs x and y are correlated because of an unobserved common cause z. In this exercise we derive a loss function to learn a random variable z such that x and y are independent given z or, equivalently, such that we have the Markov chain $x \leftarrow z \rightarrow y$.

1. Show that $I(X; Y|Z) = \mathbb{E}_{x,z \sim p(x,z)}[\mathrm{KL}(p(y|x, z) \,\|\, p(y|z))]$;

2. Using this, prove that $I(X;Y|Z) = 0$ if and only if $p(x,y|z) = p(x|z)p(y|z)$, that is, if and only if we have the Markov chain $X \leftarrow Z \rightarrow Y$ [Hint: recall the necessary condition for the KL divergence to be zero];

3. Prove that $I(X;Z|Y) = \min_{q(y|z)} \mathbb{E}_{x,z \sim p(x,z)}[\mathrm{KL}(p(y|x,z) \parallel q(y|z))]$ [Hint: it is a similar proof to what we did in class to bound $I(X;Z|Y)$]

Let $\mathcal{L}(p(y|x,z)) = \min_{q(y|z)} \mathbb{E}_{z \sim p(z)}[\mathrm{KL}(p(y|x,z) \parallel q(y|z))]$. By the previous points, x, y and z form a Markov chain if and only if $\mathcal{L}$ is minimized.

Bonus: How would you optimize $\mathcal{L}$ using a DNN?

Exercise 3 (Minimal sufficient representation, 10 points). Given a square x whose sides are colored in red and/or blue, we want to build a simple representation z for a binary classification task y where $y = 1$ if there are two consecutive sides with the same color, and $y = 0$ otherwise.

Assume all squares are equally probable, and consider the following representations:

1. $z = x$ is the identity representation;
2. $z = 1$ if the top side of the input x is red, $z = 0$ otherwise;
3. $z = \mathrm{Orb}_G(x) \in X/G$, where G is the group of planar rotations by multiples of 90 degrees (see also Exercise 2 in Assignment 1);
4. $z \in \{0, 1, 2, 3, 4\}$ is the the number of red sides of the input x.

(a) For each representation z compute the mutual information $I(z;y)$ it has with the task y (*i.e.*, how sufficient it is for the task). [Hint: Use the expression $I(y;z) = H(y) - H(y|z)$ and compute both $H(y)$ and $H(y|z)$ using the definition in the Notation section. Recall that $H(y|z) = 0$ if the value of y is completely determined by z, i.e., if y is a deterministic function of x.]

(b) For each representation z compute the information $I(x;z)$ it retains about the input (*i.e.*, how minimal it is). [Hint: use that $I(z;x) = H(z) - H(z|x)$, and notice that in all cases z is deterministic function of x.]

(c) Which representations are sufficient (that is, they maximize $I(y;z)$)? Among the sufficient representations, which one is the more minimal?

Coding assignment (optional, 15 points). In this assignment we will implement an Information Bottleneck for a reconstruction task (autoencoder), which is similar to a Variational Auto-Encoder except for the coefficient of the KL divergence term. In fact, since the task y is to reconstruct the input, we have $y = x$, in which case the Information Bottleneck Lagrangian is $\mathcal{L} = H_{p,q}(x|z) + \lambda I(z;x)$. As seen in class, we have the upper-bound $I(z;x) \leq \mathbb{E}_x \mathrm{KL}(p(z|x) \parallel q(z))$. Using this, we can rewrite the loss function as

$$\mathcal{L} = \mathbb{E}_{x \sim p(x)}\Big[\mathbb{E}_{p(z|x)}[-\log q(x|z)] + \lambda \, \mathrm{KL}(p(z|x) \parallel q(z))\Big],$$

which is the same as the same loss function used by Kingma and Welling, 2014, expect for the coefficient λ in front of the KL divergence.

1. Implement a simple standard Variational Auto-Encoder (sample implementations are also available for most frameworks)
2. Add a coefficient λ in front of the KL term in the loss function.
3. Train the modified VAE on the MNIST dataset or on the CelebA dataset[1] using different values of λ.[2]
4. For each value of λ, show the reconstruction for a few different inputs at the end of the training. How does the reconstruction change as we decrease the value of λ? You should observe that for high values of λ the VAE only roughly reconstruct some details of the input (*e.g.*, on CelebA it will reconstruct a blurry face with approximately the background color and illumination), while for lower λ it will correctly reconstructs all details.
5. Given the previous results, what information in the input image appears to be more important for the reconstruction task?
6. Would the same information be important if we change the task (for example if we were interested in a classification task)?

[1] The results on CelebA are more interesting, but you may need to use a convolutional encoder/decoder, see *e.g.* `https://github.com/keras-team/keras/blob/master/examples/variational_autoencoder_deconv.py` for a possible implementation.

[2] You may need to search the optimal range: for example, try finding first the highest value of $\lambda_{\max}$ for which you the VAE learns a non-trivial reconstruction, and then train with $\lambda \in \{\lambda_{\max}, \lambda_{\max} * 10^{-1}, \ldots, \lambda_{\max} * 10^{-n}\}$, or with a similar exponentially spaced schedule.