Incorporating mutual-information of treatment-outcome in estimating individual treatment effect

Ron Teichner

March 2020

Bottom line: There are scenarios in which a treatment is lethal for a small percentage of the population. While for the majority of the population the correlation between treatment and outcome might be positive (respond well to treatment) or zero (neutral to the treatment), for those to whom the treatment is lethal there is a high correlation between treatment and outcome. We present a method for learning an outcome estimator that is preferable in scenarios in which detecting those who are in danger by the treatment is cost-effective despite an increase in the total false-positive rate.

Contents

1 Introduction

We consider the case of estimating Individual Treatment Effect (ITE) based on observational data. Our problem formulation, assumptions and notations are identical to [Shalit et al., 2017]; hence our objective is learning a prediction model for the outcome y based on features x and treatment t. The core problem Shalit et al. were tackling was prediction errors that arise when the observational dataset is not balanced. To illustrate consider the following observational dataset:

Patients that received treatment	Patients that did not received treatment
rich: 1000 **proor**: 10	**rich**: 10 **proor**: 1000

In training the prediction error is minimized over the observational dataset. The Neural-Net is more likely to have prediction errors in regions of the input $\{x, t\}$ where the data is sparse. That is because these errors sum to a small total-prediction-error. Yet in evaluation this phenomena results in a large ITE prediction error. For estimating an ITE we must accurately predict the **two outcomes** of a rich patient receiving and not receiving treatment. We conclude that accurate prediction in the sparse input regions is as important as in the dense input regions.

In [Shalit et al., 2017] the problem is tackled by introducing a representation function of the form $\Phi : X \to R$ that learns a one-to-one transform to a new feature space R. The features represented in R are balanced in the sense that the Integral Probability Metric value, $\text{IPM}(\hat{P}_\Phi^{t=0}, \hat{P}_\Phi^{t=1})$, is small.

In the suggested project we tackle the unbalanced observational dataset problem by adding a regularization term to the prediction-loss objective. Our regularization term is maximal mutual-information of outcome and treatment given the individual patient features. If the Neural-Net fails to learn the outcome of a rich patient not receiving treatment the mutual-information will be low, thus encouraging searching an improved solution.

2 Estimating ITE: methods

In our architecture the output of the Neural-Net is a parameterized distribution $p_\theta(\hat{y} \mid t, x)$ where we denote by θ the parameters of the learned model. We propose an optimization objective that consists of minimizing the outcome prediction error and maximizing the mutual-information. Let y be the observed outcome and $\hat{y}$ the predicted outcome so that the prediction loss is $L_o(\hat{y}, y)$ and the mutual-information is $I(\hat{y}; t)$. The total loss is:

$$L = L_o(\hat{y}, y) - \gamma I(\hat{y}; t) \tag{1}$$

where γ is a hyper-parameter.

We now explicit derive $I(\hat{y}; t)$ and show that it is calculable during training:

$$\begin{aligned} \text{I}(\hat{y}; t) &= \text{H}(\hat{y}) - \text{H}(\hat{y} \mid t) \\ &= \frac{1}{N_B} \sum_x e(x) \left(a(x, 1) - \gamma(x, 1)\right) + \left(1 - e(x)\right) \left(a(x, 0) - \gamma(x, 0)\right) \end{aligned} \tag{2}$$

We defined:

$$
\begin{aligned}
a(x,t) &= \theta^{x,t} \log \sum_x \theta^{x,t} + (1-\theta^{x,t}) \log \sum_x (1-\theta^{x,t}) \\
\gamma(x,t) &= \theta^{x,t} \log \sum_x \alpha(x) + (1-\theta^{x,t}) \log \sum_x (1-\alpha(x)) \\
\alpha(x) &= e(x)\theta^{x,1} + (1-e(x))\theta^{x,0}
\end{aligned} \tag{3}
$$

The propensity score is $e(x)$ and N_B is the size of the training batch. The full derivation is at the Appendix 8. In training we estimate $\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}[\cdot]$ by sampling multiple $\hat{y}$ values at the Neural-Net output and averaging over the whole mini-batch $\{x_i, t_i\}_{i=1}^N$ of size N.

If we use negative log-likelihood as the primary objective then

$$
L_o(\hat{y}, y) = \mathrm{E}_{x,y,t\sim p_{data}(x,y,t)} [-\log p_\theta(\hat{y} = y \mid t, x)] \tag{4}
$$

Since $I(\cdot,\cdot) \geq 0$ We have that $L < L_o$. Once the model $p_\theta(\hat{y} \mid t, x)$ is trained it serves as an outcome predictor. For patient i with covariates x_i we have that $\hat{y}_i^{(1)} \sim p_\theta(\hat{y} \mid t = 1, x)$ and $\hat{y}_i^{(0)} \sim p_\theta(\hat{y} \mid t = 0, x)$.

The ITE for patient i is defined:

$$
\tau(x_i) = \mathrm{E}\left[y_i^{(1)} - y_i^{(0)} \mid x_i\right] \tag{5}
$$

The prediction of ITE using the outcome predictor:

$$
\hat{\tau}_\gamma(x_i) = \mathrm{E}_{\hat{y}_i^{(l)}\sim p_\theta(\hat{y}|t=l,x)}\left[\hat{y}_i^{(1)} - \hat{y}_i^{(0)} \mid x_i\right] \tag{6}
$$

And the prediction error is $e_\gamma(x_i) = (\tau(x_i) - \hat{\tau}_\gamma(x_i))^2$.
The prediction of ITE using the outcome predictor with $\gamma = 0$:

$$
\hat{\tau}_0(x_i) = \mathrm{E}_{\hat{y}_i^{(l)}\sim p_\phi(\hat{y}|t=l,x)}\left[\hat{y}_i^{(1)} - \hat{y}_i^{(0)} \mid x_i\right] \tag{7}
$$

The prediction error is $e_0(x_i) = (\tau(x_i) - \hat{\tau}_0(x_i))^2$.
We will be interested in cases where $e_\gamma(x_i) < e_0(x_i)$.

3 Training convergence - intuition, drawbacks

During training there are two possibilities for the prediction - either the Neural-Net will correctly predict the outcome or the Neural-Net will have a prediction error. We dwell into the different causes for these possibilities:

- Prediction error in y:
  - The Neural-Net ignores t: The Loss L_o is amplified by the low value of I. The training algorithm will search a solution in which I will have a higher value and L_o a lower one.
  - The Neural-Net did not ignore t, I has a high value that compensates the high value

of L_o. This is a drawback of the method that might be dealt with by exponentially decaying I in equation 1.

- Correct prediction of y:
  - Although the Neural-Net correctly predicts y it is motivated in increasing I which will result in wrong-prediction. To avoid this we need L_o to rise quicker than I.

4 A simple example

Consider the next example:

Idx	x	t	y	nRepetitions
1	0	1	1	N
2	1	1	0	N
3	0	0	0	N
4	1	0	1	M

We set $p_\theta(\hat{y} \mid t, x)$ as a Bernoulli distribution and we choose the model (with parameters a, b_x):

$$\begin{aligned} p_\theta(\hat{y} \mid t, x) &= \theta(t,x)\hat{y} + (1-\theta)(1-\hat{y}) \\ \theta &= \sigma(a + \mathrm{ReLU}(b_x\tilde{\tilde{x}}) + \mathrm{ReLU}(-\frac{\alpha}{b_x}\tilde{\tilde{x}})) \\ \tilde{\tilde{x}} &= \begin{bmatrix}1 & 0\end{bmatrix} R \begin{bmatrix}x-0.5 \\ t-0.5\end{bmatrix} \\ R &= \begin{bmatrix}\cos\phi & -\sin\phi \\ \sin\phi & \cos\phi\end{bmatrix}; \phi = \frac{3}{4}\pi \\ \sigma(x) &= \frac{1}{1+e^{-x}} \end{aligned} \tag{8}$$

We note that since $\hat{y}$ is binary

$$\log p_\theta(\hat{y} \mid t, x) = \log\left(\theta\hat{y} + (1-\theta)(1-\hat{y})\right) = \hat{y}\log\theta + (1-\hat{y})\log(1-\theta) \tag{9}$$

For $L_o(\hat{y}, y)$ we choose the Binary Cross Entropy loss.

$$L_o(\hat{y}, y) = -\,\mathrm{E}_{x,y,t\sim p_{data}(x,y,t)}\left[y\log\theta(t,x) + (1-y)\log(1-\theta(t,x))\right] \tag{10}$$

First let's verify that for $N = 10$ and $M = 0$ we get a perfect fit with $\gamma = 0$. What are the a, b_x, b_t values minimize $L_o^{M=0}(\hat{y}, y)$?

$$\begin{aligned} a &= -3.3539645671844482 \\ b_x &= 8.512718200683594 \end{aligned} \tag{11}$$

The θ values the model predicts:

Idx	x	t	y	θ
1	0	1	1	0.9350
2	1	1	0	0.0338
3	0	0	0	0.0338
4	1	0	1	0.0351

Indeed a perfect fit.

What a, b_x, b_t values minimize $L_o(\hat{y}, y)$ for $\alpha = 10$, $N = 3$ and $M = 1$?

$$\begin{aligned} a &= -1.8224583864212036 \\ b_x &= 6.738301753997803 \end{aligned} \tag{12}$$

The θ values the model predicts:

Idx	x	t	y	θ
1	0	1	1	0.9499
2	1	1	0	0.1391
3	0	0	0	0.1391
4	1	0	1	0.3155

Therefore the ITE is:

$$\begin{aligned} \hat{\tau}_0(x=0) &= \mathrm{E}_{\hat{y}_i^{(l)} \sim p_\phi(\hat{y}|t=l,x)} \left[\hat{y}_i^{(1)} - \hat{y}_i^{(0)} \mid x=0 \right] \\ &= \mathrm{E}_{\hat{y}_i^{(1)} \sim p_\phi(\hat{y}|t=1,x=0)} \left[\hat{y}_i^{(1)} \mid x=0 \right] - \mathrm{E}_{\hat{y}_i^{(0)} \sim p_\phi(\hat{y}|t=0,x=0)} \left[\hat{y}_i^{(0)} \mid x=0 \right] \\ &= 0.9499 - 0.1391 = 0.8108 \\ \hat{\tau}_0(x=1) &= \mathrm{E}_{\hat{y}_i^{(l)} \sim p_\phi(\hat{y}|t=l,x)} \left[\hat{y}_i^{(1)} - \hat{y}_i^{(0)} \mid x=1 \right] \\ &= \mathrm{E}_{\hat{y}_i^{(1)} \sim p_\phi(\hat{y}|t=1,x=1)} \left[\hat{y}_i^{(1)} \mid x=1 \right] - \mathrm{E}_{\hat{y}_i^{(0)} \sim p_\phi(\hat{y}|t=0,x=1)} \left[\hat{y}_i^{(0)} \mid x=1 \right] \\ &= 0.1391 - 0.3155 = -0.1764 \end{aligned} \tag{13}$$

While the true ITE is:

$$\begin{aligned} \tau(x=0) &= 1 \\ \tau(x=1) &= -1 \end{aligned} \tag{14}$$

The ITE percentage error per x value is:

$$\begin{aligned} err(x=0) &= 100 \frac{1 - 0.8108}{1} = 18.92\% \\ err(x=1) &= 100 \frac{-1 - (-0.1764)}{-1} = 82.36\% \end{aligned} \tag{15}$$

The θ values the model predicts for $\gamma = 0.55$:

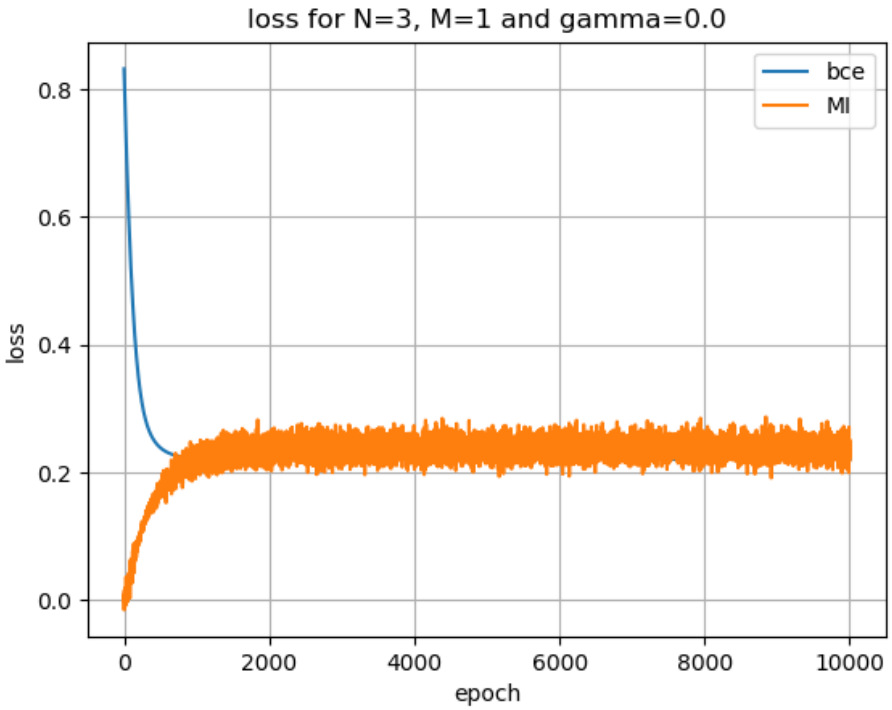

Figure 1: BCE loss & mutual-info

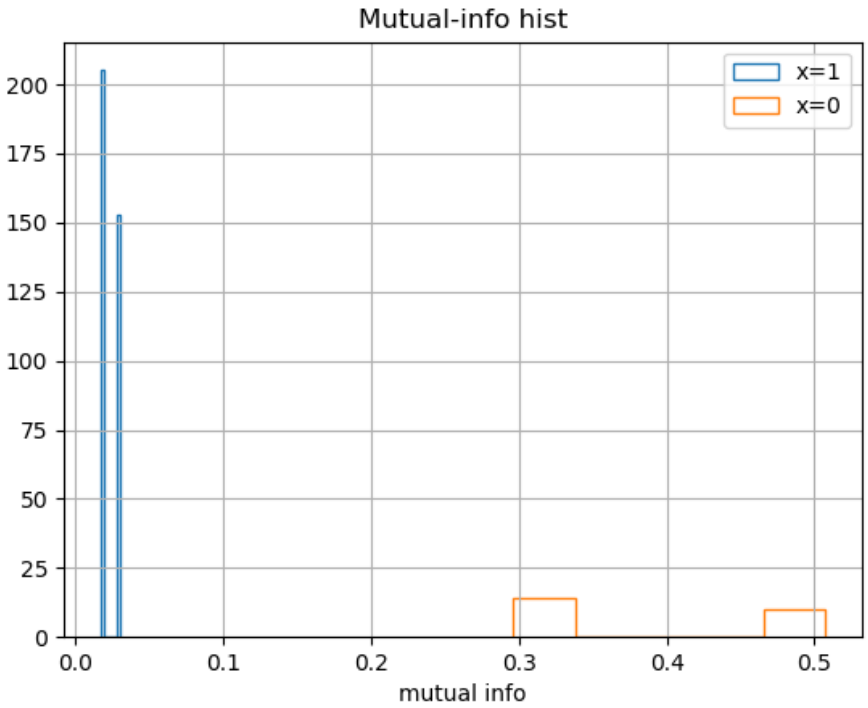

Figure 2: mutual-info-hist, $\gamma = 0$

Idx	x	t	y	θ
1	0	1	1	0.8656
2	1	1	0	0.1715
3	0	0	0	0.1715
4	1	0	1	0.4692

Therefore the ITE is:

$$\begin{aligned}
\hat{\tau}_0(x=0) &= \mathrm{E}_{\hat{y}_i^{(l)} \sim p_\phi(\hat{y}|t=l,x)} \left[\hat{y}_i^{(1)} - \hat{y}_i^{(0)} \mid x=0 \right] \\
&= \mathrm{E}_{\hat{y}_i^{(1)} \sim p_\phi(\hat{y}|t=1,x=0)} \left[\hat{y}_i^{(1)} \mid x=0 \right] - \mathrm{E}_{\hat{y}_i^{(0)} \sim p_\phi(\hat{y}|t=0,x=0)} \left[\hat{y}_i^{(0)} \mid x=0 \right] \\
&= 0.8656 - 0.1715 = 0.6941 \\
\hat{\tau}_0(x=1) &= \mathrm{E}_{\hat{y}_i^{(l)} \sim p_\phi(\hat{y}|t=l,x)} \left[\hat{y}_i^{(1)} - \hat{y}_i^{(0)} \mid x=1 \right] \\
&= \mathrm{E}_{\hat{y}_i^{(1)} \sim p_\phi(\hat{y}|t=1,x=1)} \left[\hat{y}_i^{(1)} \mid x=1 \right] - \mathrm{E}_{\hat{y}_i^{(0)} \sim p_\phi(\hat{y}|t=0,x=1)} \left[\hat{y}_i^{(0)} \mid x=1 \right] \\
&= 0.1715 - 0.4692 = -0.29769
\end{aligned} \tag{16}$$

While the true ITE is:

$$\begin{aligned}
\tau(x=0) &= 1 \\
\tau(x=1) &= -1
\end{aligned} \tag{17}$$

The ITE percentage error per x value is:

$$\begin{aligned}
err(x=0) &= 100 \frac{1 - 0.6941}{1} = 30.58\% \\
err(x=1) &= 100 \frac{-1 - (-0.29769)}{-1} = 70.23\%
\end{aligned} \tag{18}$$

We repeat all the results in the next two tables and conclude that the model trained with the hybrid BCE-mutual-information loss has improved the ITE for $x = 1$ while increasing the error for $x = 0$. This is due to the inherent trade-off between the two arguments in the hybrid loss.

Model predictions:

Idx	x	t	y	$p_\theta(\hat{y}=1 \mid x,t); \gamma=0$	$p_\theta(\hat{y}=1 \mid x,t); \gamma=0.55$
1	0	1	1	0.9499	0.8656
2	1	1	0	0.1391	0.1715
3	0	0	0	0.1391	0.1715
4	1	0	1	0.3155	0.4692

ITE errors:

x	ITE(x=0) error; $\gamma=0$	ITE(x=1) error; $\gamma=0.55$
0	19%	30%
1	82%	70%

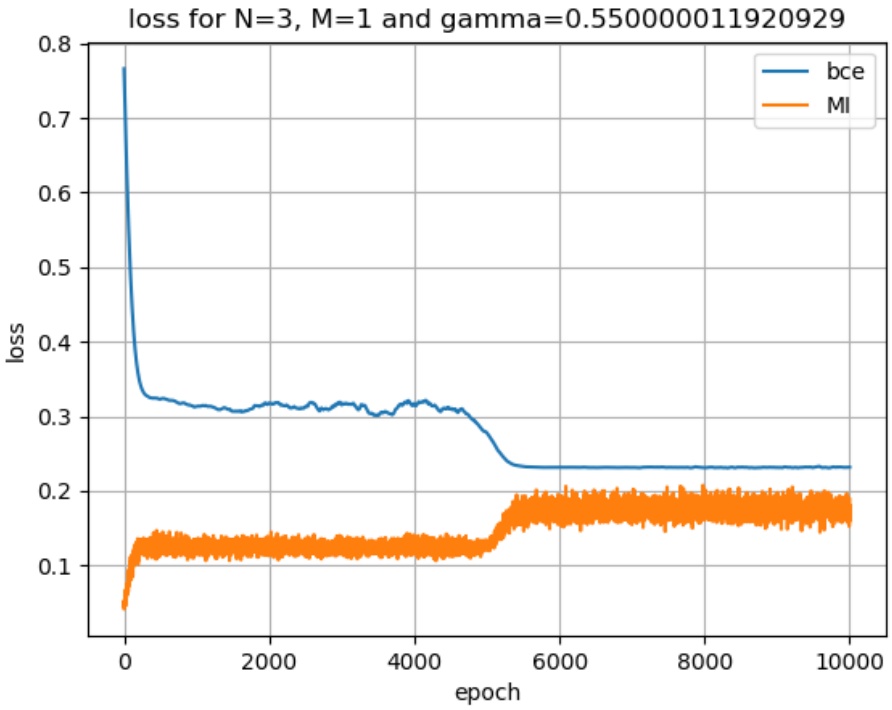

Figure 3: BCE loss & mutual-info

5 Theoretical analysis

We would like to examine the change in log-likelihood and mutual-information due to an update to the learned model. They both will increase if **a.** the estimated-outcome error decreases and **b.** the estimated-outcome values become more extreme (Bernoullii distribution tends to 0,1 for a binary outcome). They contradict if the estimated-outcome values become more extreme on expense of an increase in the estimated-outcome error.

We numerically examined the change due an update to the learned-model. Without loss of generality, we focus on an update which changes only the probability of patients with covariates x_0 that are not treated - $p(\hat{y} = 1 \mid x_0, t = 0)$. For these patients there is a 30% chance for a positive outcome when not receiving the treatment; Therefore the log-likelihood of the model should peak at $p(\hat{y} = 1 \mid x_0, t = 0) = 0.3$. We analyze the change in log-likelihood and mutual information for an infinitesimal update that decreases the probability $p(\hat{y} = 1 \mid x_0, t = 0)$ by 0.001.

In figure 4 we plotted the log-likelihood and mutual-information values contributed by patients with covariates x_0 and different propensity scores. At the lower-left figure we see, as expected, that the log-likelihood value peaks when $p(\hat{y} = 1 \mid x_0, t = 0) = 0.3$, yet, the mutual-information value, seen at the lower-right figure, peaks at $p(\hat{y} = 1 \mid x_0, t = 0) = 0$. At the upper figures we see that if the leaned-model predicts $p(\hat{y} = 1 \mid x_0, t = 0) > 0.3$ than the infinitesimal decrease improves (increases) both the log-likelihood and the mutual information. But, if the leaned-model predicts $p(\hat{y} = 1 \mid x_0, t = 0) < 0.3$ than this infinitesimal decrease increases the mutual-information while decreasing the log-likelihood.

The drift away from the correct output distribution of $p(\hat{y} = 1 \mid x_0, t = 0) = 0.3$ de-

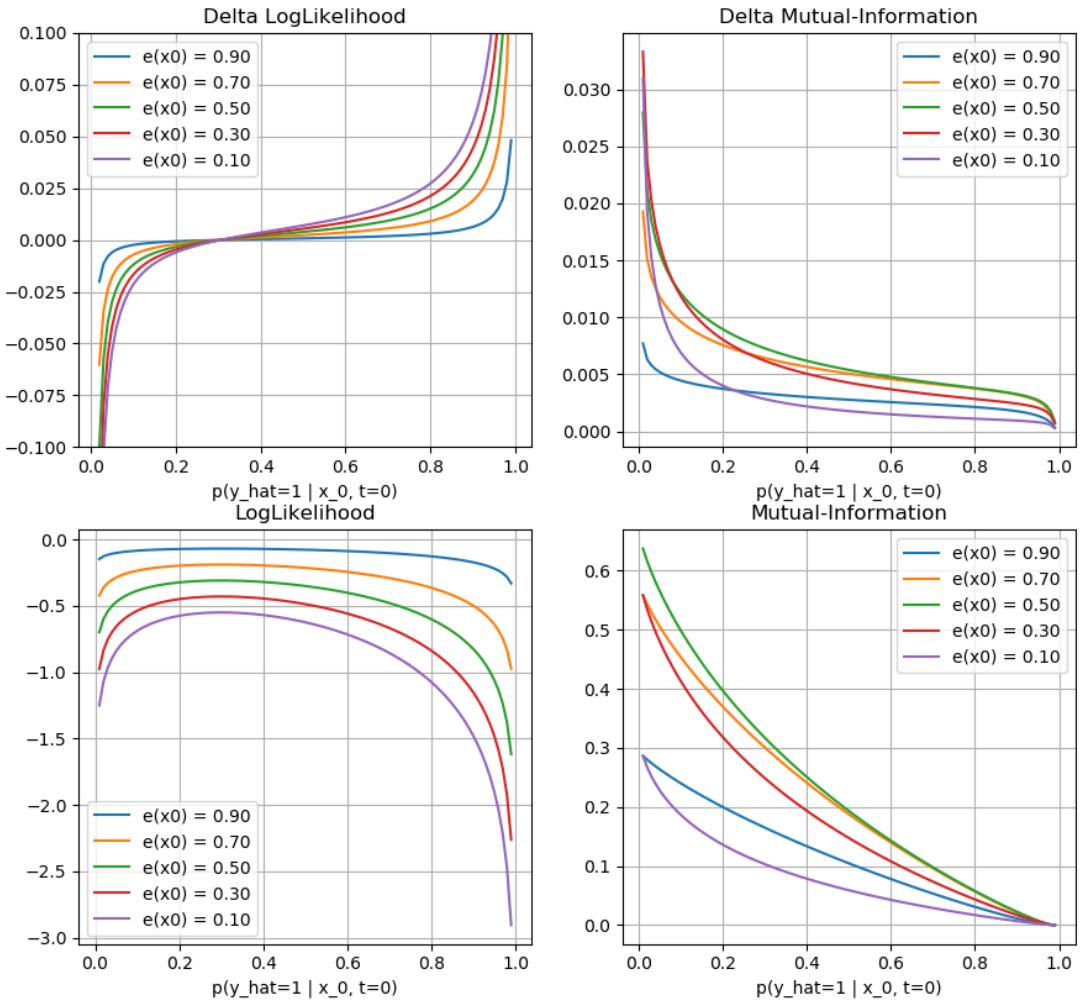

Figure 4: Log-likelihood and mutual-information values. In the upper figures the change due to an infinitesimal update to the learned mode; In the lower figures the values of log-likelihood and mutual-information

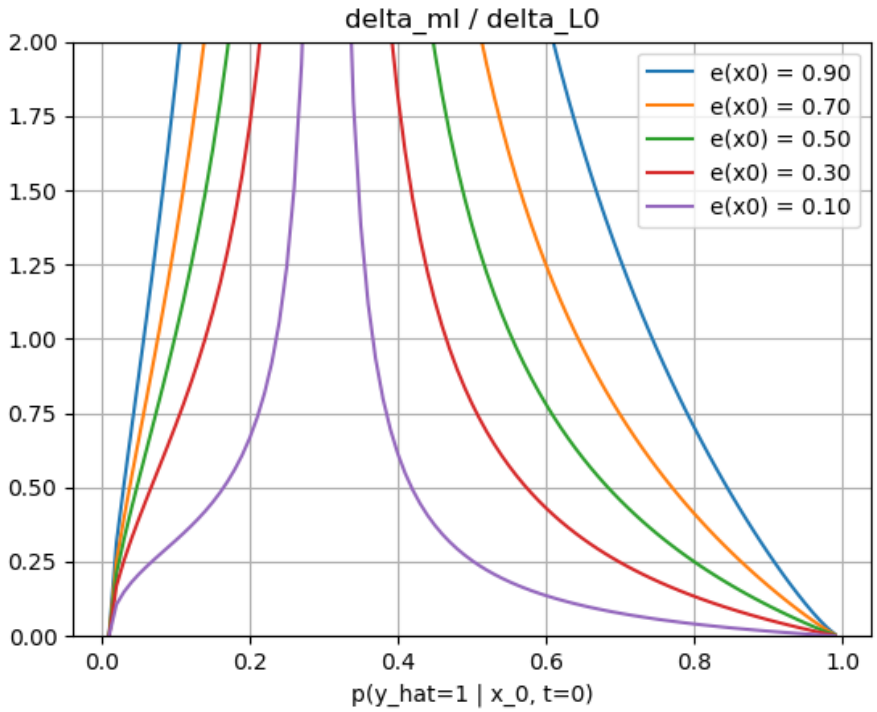

Figure 5: Ratio of change in mutual-information to change in log-likelihood due to an update of the learned model for specific covariates. Different propensity scores are displayed.

pends on the value of the propensity score of x_0 as can be seen in figure 5. The learned value of $p(\hat{y} = 1 \mid x_0, t = 0)$ will drift away from 0.3 towards 0 as long as the increase in mutual-information is higher than the decrease in log-likelihhod. We can see that for patients with low propensity scores the drift is smaller. This is potentially a desired feature of the proposed method. The model that is trained only with respect to log-likelihood is prone to mistake for pairs of (covariates, treatment) that are rare. We focus here on the model-estimation when $t = 0$ and note that when $t = 0$ is rare (high propensity score) we get a larger region in which the mutual-information is greater than the log-likelihood. This amplifies the attraction towards $p(\hat{y} = 1 \mid x_0, t = 0) = 0$ - a desired feature if at the current train-iteration $p(\hat{y} = 1 \mid x_0, t = 0) > 0.3$ and undesired if $p(\hat{y} = 1 \mid x_0, t = 0) < 0.3$.

We conclude (based just on an 'healthy engineering feel') that the proposed method will have in average a higher outcome estimation error. That is because most estimated outcomes will drift apart from their true values due to the mutual-information loss. Only in specific scenarios and on specific, more rare populations, it might have a lower outcome estimation error. What scenarios exactly do we mean? In many cases the majority of patients respond well to a treatment or are neutral to the treatment. Yet there exists a small fraction of the patients to which the treatment is lethal. We would like to be able to better target these patients even at the cost of a higher false-positive rate. The same concept applies also if some patients have a rare set of covariates and are in great danger pre-treatment assignment.

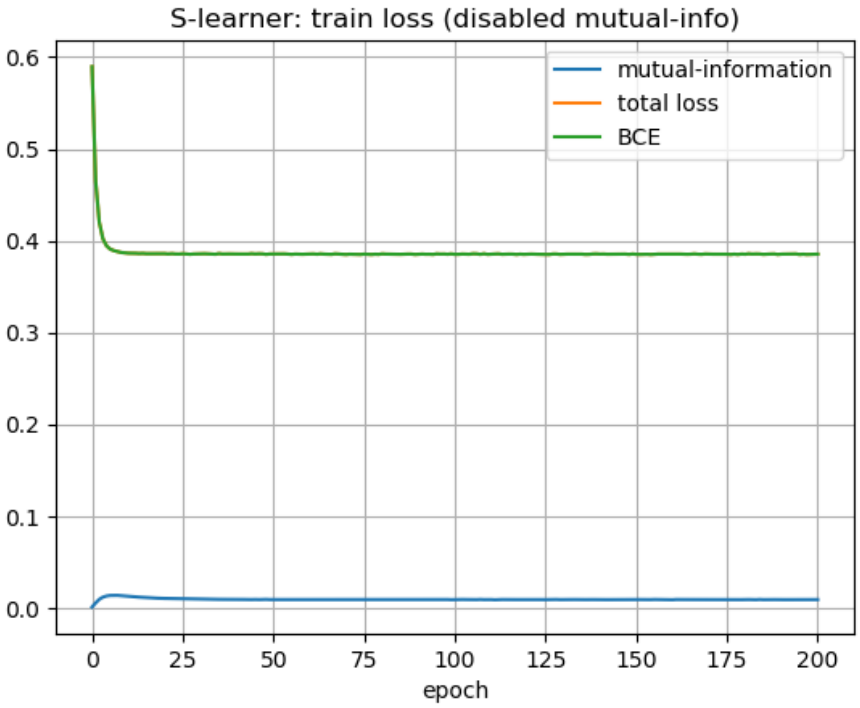

Figure 6: Train with respect to Binary-Cross-Entropy loss. Mutual-information not included in the loss objective

6 Twins dataset

The twins dataset is a dataset on twin births, adapted from NBER and manipulated to emulate an observational study. It deals with twins born at a low absolute-weight and considers the treatment as being born with the higher weight. Assuming all covariates for the twins are identical we have here a rare case in which we have both the factual and counterfactual outcomes available. Therefore ITE can be estimated and compared to the true ITE values. See section 4.3 in [Louizos et al., 2017] for further details.

We trained[1] an S-learner [Künzel et al., 2019] which is a single learned-model that views the treatment as just another input on top of the covariates. Our model is a simple logistic-regression model - a sigmoid activation function that is applied to an affine transformation of the input. The outcome at the Twins dataset is binary (live or death) and therefore our model outputs the Bernoulli distribution parameter $p(\hat{y} = 1 \mid x, t)$. The nominal-train is with respect to the Binary Cross-Entropy (BCE) loss function. We then trained the model with respect to the hybrid BCE - mutual-information loss.

Figures 6 and 7 depicts the BCE loss, the mutual-information and the total loss when training with respect to BCE loss only (figure 6) and when training with respect to the hybrid loss (figure 7). We can clearly see the result of the losses being in trade-off. While the BCE loss, when training with respect to BCE loss only stabilizes at a value of almost 0.4, when training with respect to the hybrid loss it stabilizes at a value a-bit higher than 0.4.

[1]Code is available at https://github.com/RonTeichner/Benchmarks

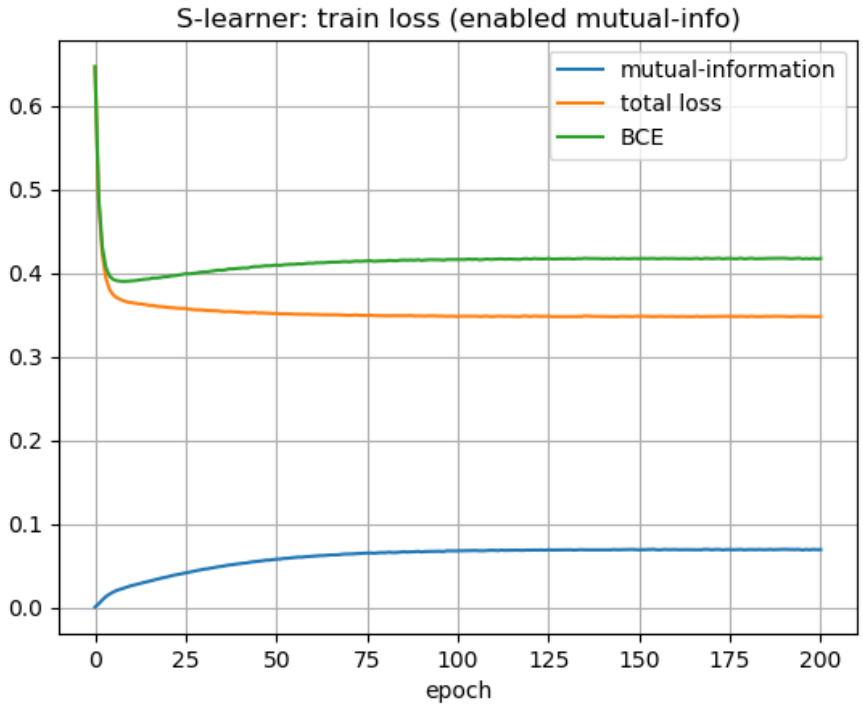

Figure 7: Train with respect to hybrid Binary-Cross-Entropy mutual-information loss

Type	BCE-loss	Hybrid-loss	Ground-truth
Light twin	14.97	46.35	100
Heavy twin	3.9	0.65	0

Table 1: First group: twins in which the lighter baby does not survive (6.4% of the dataset). Mortality percentage by hard-decision

We analyzed the results for 3 populations. The first consists of twins in which the heavier baby survived and the lighter did not (6.4% of the dataset); the second consists of twins in which the heavier baby did not survive and the lighter did survive (3.9% of the dataset); the third group consists of twins that both either survived or not survived (89.7% of the dataset). For each individual we view the model prediction of mortality probability. We translate the mortality probability into hard-decisions - if it is higher than 0.5 we say the model predicts mortality. The results are depicted in tables 1 - 3. The results on average show a great increase in the estimation outcome error.

- The largest group are twins in which both twins survive (76.5% of the dataset). While the model that it trained with respect to BCE loss only predicts that 2.6% of them will not survive, the model trained on the hybrid loss predicts that 13.6% of them will not survive. This is an increase in the rate of false-positive.
- The second largest group (12.46%) are twins in which both twins do not survive. In this group we see that the model trained on the hybrid loss predicted that 39.44% will not survive while the model trained on BCE only predicted that only 20.5% will not survive.
- The third largest group (6.4%) are twins in which the lighter twin does not survive and the heavier twin does survive. In this group we see that while the BCE-only model predicts

Type	BCE-loss	Hybrid-loss	Ground-truth
Light twin	15.2	54.3	0
Heavy twin	3	0.43	100

Table 2: Second group: twins in which the heavier baby does not survive (3.9% of the dataset). Mortality percentage by hard-decision

Type	BCE-loss	Hybrid-loss	Ground-truth	% in group
Twins survive	2.6	13.67	0	86
Twins not survive	20.57	39.4	100	14

Table 3: Third group: twins in which both either survive or not (89.7% of the dataset). Mortality percentage by hard-decision

that only 15% of lighter babies will not survive, the hybrid model predicts that 46.35% of lighter babies will not survive. This is an improvement, but again we see an increase in false-positive because the hybrid model predicts that 3.9% of heavier babies will not survive in comparison to 0.6% according to the BCE only model.

- In the last and smallest group (3.99%) we obtained undesired results - a decrease in the no. of estimated heavy babies that will not survive (hybrid model vs BCE).

7 Conclusions and future work

We introduced a hybrid loss that on top of a nominal log-likelihood term contains regularization term that encourages high mutual-information between treatment and outcome. We showed on a simple example (Section 4) that the hybrid loss does amplify the loss for rare inputs in the dataset. This amplification resulted in a decrease of the ITE error for these rare inputs but came at a cost of increasing the ITE error of the majority of inputs. The same phenomena was observed on analysis of results obtained on the Twins dataset (Section 6.

We argue that for scenarios in which there is a health risking rare outcome, a rise in the rate of false-positives is acceptable if coupled to an increase in detection performance of the patients in greater danger. For future work we suggest combining the estimated outcome from two models - one trained with log-likelihood loss only and the other with the hybrid loss. We suggest a combination that weights both the probability of the set of covariates of a patient, x and the propensity score $e(x)$ for combining the two estimations. By identifying the rare patients we could assign them the estimation of the hybrid model which hopefully better fits rare inputs.

References

[Künzel et al., 2019] Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165.

[Louizos et al., 2017] Louizos, C., Shalit, U., Mooij, J. M., Sontag, D., Zemel, R., and Welling, M. (2017). Causal effect inference with deep latent-variable models. In Guyon, I., Luxburg, U. V., Bengio, S., Wallach, H., Fergus, R., Vishwanathan, S., and Garnett, R., editors, *Advances in Neural Information Processing Systems 30*, pages 6446–6456. Curran Associates, Inc.

[Shalit et al., 2017] Shalit, U., Johansson, F. D., and Sontag, D. (2017). Estimating individual treatment effect: generalization bounds and algorithms. In Precup, D. and Teh, Y. W., editors, *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 3076–3085, International Convention Centre, Sydney, Australia. PMLR.

8 Appendix

Mutual information derivation:

$$\mathrm{I}(\hat{y};t) = \mathrm{H}(\hat{y}) - \mathrm{H}(\hat{y} \mid t) \tag{19}$$

$$\begin{aligned}
\mathrm{H}(\hat{y}) &= -\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y})}\left[\log p_\theta(\hat{y})\right] \\
&= -\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y})}\left[\log \sum_{x,t} p_\theta(\hat{y},x,t)\right] \\
&= -\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y})}\left[\log \sum_{x,t} p_\theta(\hat{y}\mid x,t)p_{data}(x,t)\right] \\
&= -\,\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \sum_{x,t} p_\theta(\hat{y}\mid x,t)p_{data}(x,t)\right] \\
&= -\,\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \sum_{x} p_{data}(x)\sum_{t} p_\theta(\hat{y}\mid x,t)p_{data}(t\mid x)\right] \\
&= -\,\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \frac{1}{N_B}\sum_{x}\sum_{t} p_\theta(\hat{y}\mid x,t)p_{data}(t\mid x)\right] \\
&= -\,\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \frac{1}{N_B}\sum_{x} e(x)p_\theta(\hat{y}\mid x,1) + (1-e(x))p_\theta(\hat{y}\mid x,0)\right]
\end{aligned} \tag{20}$$

Develop $\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \frac{1}{N_B}\sum_x e(x)p_\theta(\hat{y}\mid x,1) + (1-e(x))p_\theta(\hat{y}\mid x,0)\right]$:

$$\begin{aligned}
&\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \frac{1}{N_B}\sum_{x} e(x)p_\theta(\hat{y}\mid x,1)(1-e(x))p_\theta(\hat{y}\mid x,0)\right] \\
&= \theta^{x,t}\left[\log \frac{1}{N_B}\sum_{x} e(x)\theta^{x,1} + (1-e(x))\theta^{x,0}\right] \\
&+ (1-\theta^{x,t})\left[\log \frac{1}{N_B}\sum_{x} e(x)(1-\theta^{x,1}) + (1-e(x))(1-\theta^{x,0})\right]
\end{aligned} \tag{21}$$

Develop $\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\frac{1}{N_B}\sum_x e(x)p_\theta(\hat{y}\mid x,1)+(1-e(x))p_\theta(\hat{y}\mid x,0)\right]$:

$$
\begin{aligned}
&\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\frac{1}{N_B}\sum_x e(x)p_\theta(\hat{y}\mid x,1)+(1-e(x))p_\theta(\hat{y}\mid x,0)\right]\\
&=\mathrm{E}_{t\sim p_{data}(t|x)}(\theta^{x,t}\left[\log\frac{1}{N_B}\sum_x e(x)\theta^{x,1}+(1-e(x))\theta^{x,0}\right]\\
&+(1-\theta^{x,t})\left[\log\frac{1}{N_B}\sum_x e(x)(1-\theta^{x,1})+(1-e(x))(1-\theta^{x,0})\right])\\
&=e(x)(\theta^{x,1}\left[\log\frac{1}{N_B}\sum_x e(x)\theta^{x,1}+(1-e(x))\theta^{x,0}\right]\\
&+(1-\theta^{x,1})\left[\log\frac{1}{N_B}\sum_x e(x)(1-\theta^{x,1})+(1-e(x))(1-\theta^{x,0})\right])\\
&+(1-e(x))(\theta^{x,0}\left[\log\frac{1}{N_B}\sum_x e(x)\theta^{x,1}+(1-e(x))\theta^{x,0}\right]\\
&+(1-\theta^{x,0})\left[\log\frac{1}{N_B}\sum_x e(x)(1-\theta^{x,1})+(1-e(x))(1-\theta^{x,0})\right])\\
&=\log\frac{1}{N_B}+e(x)\left(\theta^{x,1}\log\sum_x\alpha(x)+(1-\theta^{x,1})\log\sum_x\beta(x)\right)\\
&+(1-e(x))\left(\theta^{x,0}\log\sum_x\alpha(x)+(1-\theta^{x,0})\log\sum_x\beta(x)\right)\\
&=\log\frac{1}{N_B}+e(x)\gamma(x,1)+(1-e(x))\gamma(x,0)
\end{aligned}
\tag{22}
$$

We defined:

$$
\begin{aligned}
\gamma(x,t)&=\theta^{x,t}\log\sum_x\alpha(x)+(1-\theta^{x,t})\log\sum_x\beta(x)\\
&=\theta^{x,t}\log\sum_x\alpha(x)+(1-\theta^{x,t})\log\sum_x(1-\alpha(x))\\
\alpha(x)&=e(x)\theta^{x,1}+(1-e(x))\theta^{x,0}\\
\beta(x)&=e(x)(1-\theta^{x,1})+(1-e(x))(1-\theta^{x,0})=1-\alpha(x)
\end{aligned}
\tag{23}
$$

Develop $\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \frac{1}{N_B}\sum_x e(x)p_\theta(\hat{y}\mid x,1)+(1-e(x))p_\theta(\hat{y}\mid x,0)\right]$:

$$\begin{aligned}
&\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log \frac{1}{N_B}\sum_x e(x)p_\theta(\hat{y}\mid x,1)+(1-e(x))p_\theta(\hat{y}\mid x,0)\right]\\
&=\mathrm{E}_{x\sim p_{data}(x)}\left(\log\frac{1}{N_B}+e(x)\gamma(x,1)+(1-e(x))\gamma(x,0)\right)\\
&=\frac{1}{N_B}\sum_x\left(\log\frac{1}{N_B}+e(x)\gamma(x,1)+(1-e(x))\gamma(x,0)\right)\\
&=\log\frac{1}{N_B}+\frac{1}{N_B}\sum_x\left(e(x)\gamma(x,1)+(1-e(x))\gamma(x,0)\right)
\end{aligned}\tag{24}$$

To conclude:

$$\mathrm{H}(\hat{y})=-\log\frac{1}{N_B}-\frac{1}{N_B}\sum_x\left(e(x)\gamma(x,1)+(1-e(x))\gamma(x,0)\right)\tag{25}$$

Now develop $\mathrm{H}(\hat{y}\mid t)$:

$$\begin{aligned}
\mathrm{H}(\hat{y}\mid t)&=-\mathrm{E}_{\hat{y},t\sim p_\theta(\hat{y},t)}\left[\log p_\theta(\hat{y}\mid t)\right]\\
&=-\mathrm{E}_{\hat{y},t\sim p_\theta(\hat{y},t)}\left[\log\sum_x p_\theta(\hat{y},x\mid t)\right]\\
&=-\mathrm{E}_{\hat{y},t\sim p_\theta(\hat{y},t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]\\
&=-\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]
\end{aligned}\tag{26}$$

Develop $\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]$:

$$\begin{aligned}
&\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]\\
&=\theta^{x,t}\log\frac{1}{N_B}\sum_x\theta^{x,t}+(1-\theta^{x,t})\log\frac{1}{N_B}\sum_x(1-\theta^{x,t})\\
&=\log\frac{1}{N_B}+\theta^{x,t}\log\sum_x\theta^{x,t}+(1-\theta^{x,t})\log\sum_x(1-\theta^{x,t})
\end{aligned}\tag{27}$$

Develop $\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]$:

$$
\begin{aligned}
&\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]\\
&=\mathrm{E}_{t\sim p_{data}(t|x)}\left(\log\frac{1}{N_B}+\theta^{x,t}\log\sum_x\theta^{x,t}+(1-\theta^{x,t})\log\sum_x(1-\theta^{x,t})\right)\\
&=e(x)\left(\log\frac{1}{N_B}+\theta^{x,1}\log\sum_x\theta^{x,1}+(1-\theta^{x,1})\log\sum_x(1-\theta^{x,1})\right)\\
&+(1-e(x))\left(\log\frac{1}{N_B}+\theta^{x,0}\log\sum_x\theta^{x,0}+(1-\theta^{x,0})\log\sum_x(1-\theta^{x,0})\right)\\
&=\log\frac{1}{N_B}+e(x)\left(\theta^{x,1}\log\sum_x\theta^{x,1}+(1-\theta^{x,1})\log\sum_x(1-\theta^{x,1})\right)\\
&+(1-e(x))\left(\theta^{x,0}\log\sum_x\theta^{x,0}+(1-\theta^{x,0})\log\sum_x(1-\theta^{x,0})\right)\\
&=\log\frac{1}{N_B}+e(x)a(x,1)+(1-e(x))a(x,0)
\end{aligned}
\tag{28}
$$

We define:

$$
a(x,t)=\theta^{x,t}\log\sum_x\theta^{x,t}+(1-\theta^{x,t})\log\sum_x(1-\theta^{x,t})
\tag{29}
$$

Develop $\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]$:

$$
\begin{aligned}
&\mathrm{E}_{x\sim p_{data}(x)}\,\mathrm{E}_{t\sim p_{data}(t|x)}\,\mathrm{E}_{\hat{y}\sim p_\theta(\hat{y}|x,t)}\left[\log\sum_x p_{data}(x)p_\theta(\hat{y}\mid x,t)\right]\\
&=\mathrm{E}_{x\sim p_{data}(x)}\left(\log\frac{1}{N_B}+e(x)a(x,1)+(1-e(x))a(x,0)\right)\\
&=\frac{1}{N_B}\sum_x\left(\log\frac{1}{N_B}+e(x)a(x,1)+(1-e(x))a(x,0)\right)\\
&=\mathrm{E}_{x\sim p_{data}(x)}\left(\log\frac{1}{N_B}+e(x)a(x,1)+(1-e(x))a(x,0)\right)\\
&=\log\frac{1}{N_B}+\frac{1}{N_B}\sum_x\left(e(x)a(x,1)+(1-e(x))a(x,0)\right)
\end{aligned}
\tag{30}
$$

To conclude:

$$
\mathrm{H}(\hat{y}\mid t)=-\log\frac{1}{N_B}-\frac{1}{N_B}\sum_x\left(e(x)a(x,1)+(1-e(x))a(x,0)\right)
\tag{31}
$$

So,

$$\begin{aligned}
\mathrm{I}(\hat{y};t) &= \mathrm{H}(\hat{y}) - \mathrm{H}(\hat{y} \mid t) \\
&= -\log\frac{1}{N_B} - \frac{1}{N_B}\sum_x \left(e(x)\gamma(x,1) + (1-e(x))\gamma(x,0)\right) \\
&+ \log\frac{1}{N_B} + \frac{1}{N_B}\sum_x \left(e(x)a(x,1) + (1-e(x))a(x,0)\right) \\
&= \frac{1}{N_B}\sum_x e(x)\left(a(x,1) - \gamma(x,1)\right) + (1-e(x))\left(a(x,0) - \gamma(x,0)\right)
\end{aligned} \tag{32}$$

We defined:

$$\begin{aligned}
a(x,t) &= \theta^{x,t}\log\sum_x \theta^{x,t} + (1-\theta^{x,t})\log\sum_x (1-\theta^{x,t}) \\
\gamma(x,t) &= \theta^{x,t}\log\sum_x \alpha(x) + (1-\theta^{x,t})\log\sum_x (1-\alpha(x)) \\
\alpha(x) &= e(x)\theta^{x,1} + (1-e(x))\theta^{x,0}
\end{aligned} \tag{33}$$

Some results for expectations:

$$\begin{aligned}
&\mathrm{E}_{x,y,t\sim p(x,y,t)}\left[f(x,y,t)\right] \\
&= \int_{x,y,t} p(x,y,t)f(x,y,t)dxdydt \\
&= \int_{x,y} p(y,x)\left[\int_t p(t \mid y,x)f(x,y,t)dt\right]dxdy \\
&= \mathrm{E}_{x,y\sim p(x,y)}\left[\mathrm{E}_{t\sim p(t|y,x)}\left[f(x,y,t)\right]\right]
\end{aligned}$$

$$\begin{aligned}
&\mathrm{E}_{t\sim p(t)}\left[f(t)\right] \\
&= \int_t f(t)p(t)dt \\
&= \int_t f(t)\left[\int_x p(t \mid x)p(x)dx\right]dt \\
&= \int_x p(x)dx\int_t f(t)p(t \mid x)dt \\
&= \mathrm{E}_{x\sim p(x)}\left[\mathrm{E}_{t\sim p(t|x)}\left[f(t)\right]\right]
\end{aligned}$$