Advanced Machine Learning

Jaime Ferrando Huertas

May 2021

1 Exercise

Lets begin explaining what $D(p(x)\|q(x))$ is:

$$D(p(x)\|q(x)) = \sum_{x\in X} p(x) \ln \frac{p(x)}{q(x)} \tag{1}$$

This term is a measure of the difference between two probability distributions over the same variable x, also called *Kullback-Leibler divergence*. This is a non-symmetric measure of the difference between two probability distributions p(x) and q(x).

To demonstrate is not symetric we need to prove:

$$D(p(x)\|q(x)) \neq D(q(x)\|p(x)) \tag{2}$$

Lets propose p and q as follows:

$$\begin{array}{l} P : X \sim Binomial(2, 0.5) \\ Q : X \sim Uniform(0, 2) \end{array} \tag{3}$$

With p and q:

$$p(x) = \begin{cases} 1/4, & \text{if } x = 0 \\ 1/2, & \text{if } x = 1 \\ 1/4, & \text{if } x = 2 \end{cases} \tag{4}$$

$$q(x) = \frac{1}{3} \tag{5}$$

We can calculate the exact result now.

$$\begin{aligned} D(p(x)\|q(x)) &= \sum_{x\in\mathcal{X}} p(x) \cdot \log \frac{p(x)}{q(x)} \\ &= \frac{1}{4}\log\frac{3}{4} + \frac{1}{2}\log\frac{3}{2} + \frac{1}{4}\log\frac{3}{4} \\ &= \frac{1}{2}\log\frac{9}{8} = 0.0589 \end{aligned} \tag{6}$$

$$\begin{aligned} D(q(x)\|p(x)) &= \sum_{x\in\mathcal{X}} q(x) \cdot \log \frac{q(x)}{p(x)} \\ &= \frac{1}{3}\log\frac{4}{3} + \frac{1}{3}\log\frac{2}{3} + \frac{1}{3}\log\frac{4}{3} \\ &= \frac{1}{3}\log\frac{32}{27} = 0.0566 \end{aligned} \tag{7}$$

We have found a non symetric case. To prove that it doesn't satisfy the trianqle equaliy we can introduce a new distribution u and need to find a case that doesn't satisfy:

$$D(p(x)||q(x)) <= D(p(x)||u(x)) + D(u(x)||q(x)) \tag{8}$$

Let's define u as:

$$u(x) = \begin{cases} 0.36, & \text{if } x = 0 \\ 0.36, & \text{if } x = 1 \\ 0.28, & \text{if } x = 2 \end{cases} \tag{9}$$

We can now calculate:

$$D(p(x)\|u(x)) = 0.0448 \tag{10}$$

$$D(u(x)\|q(x)) = 0.0066 \tag{11}$$

And the triangle equality does not hold.

$$\begin{array}{ll} D(p(x)||q(x)) & <= D(p(x)||u(x)) + D(u(x)||q(x)) \\ 0.0566 & <= 0.0448 + 0.0066 \\ 0.0566 & <= 0.0514 \\ & \text{Does not hold!} \end{array} \tag{12}$$

2 Exercise

Yes, $H(X \mid Y = y) \geq H(X)$ is possible. To demonstrate lets first take some definitions from the slides.

$$\begin{aligned} H(X) &= -\sum_x p(x) \log p(x) \\ H(X,Y) &= H(X) + H(Y \mid X) \\ H(Y \mid X) &= \sum_x p(x) H(Y \mid X = x) \\ &= -\sum_x \sum_y p(x,y) \log p(y \mid x) \end{aligned} \tag{13}$$

We can propose an example where the hypothesis holds. Let (X,Y) have the following join distribution:

$Y \backslash X$	1	2	$\sum$
1	0	3/4	3/4
2	1/8	1/8	1/4
$\sum$	1/8	7/8	1

(14)

Where we have

$$\begin{aligned} H(X) &= H\left(\frac{1}{8}, \frac{7}{8}\right) = 0.544 bits \\ H(X|Y=1) &= 0 bits \\ H(X|Y=2) &= 1 bits \end{aligned} \tag{15}$$

We have found a case where $H(X \mid Y = y) \geq H(X)$, specifically $H(X \mid Y = 2) \geq H(X)$. Note that the logs are base two.

3 Exercise

In this exercise we follow exercise in slides 22 with a new string "abaaaa" of length 6 and "aba" as a prefix. To compute the table we follow equations in slide 21.

$$\text{For } 0 \leq j < |Q|. \qquad \begin{array}{r} H_0(j) = 0 \\ c_0(j) = I(j) \end{array} \tag{16}$$

Recursion. For $0 \leq j < |\mathcal{Q}| - 1; 1 \leq t \leq |w|$:

$$\begin{aligned} c_t(j) &= \frac{\sum_{i=0}^{|Q|-1} c_{t-1}(i) P\left(i, w_t, j\right)}{\sum_{k=0}^{|Q|-1} \sum_{i=0}^{|Q|-1} c_{t-1}(i) P\left(i, w_t, k\right)} \\ p\left(\theta_{t-1} = i \mid \theta_t = j, w_{1,t}\right) &= \frac{c_{t-1}(i) P\left(i, w_t, j\right)}{\sum_{k=0}^{|\mathcal{Q}|-1} c_{t-1}(k) P\left(k, w_t, j\right)} \\ H_t(j) &= \sum_{i=0}^{|Q|-1} H_{t-1}(i) p\left(\theta_{t-1} = i \mid \theta_t = j, w_{1,t}\right) \\ &\quad - \sum_{i=0}^{|\mathcal{Q}|-1} p\left(\theta_{t-1} = i \mid \theta_t = j, w_{1,t}\right) \log p\left(\theta_{t-1} = i \mid \theta_t = j, w_{1,t}\right) \end{aligned} \tag{17}$$

Termination: For $0 \leq j < |Q| - 1; T = |\omega| + 1$:

$$\begin{aligned} &H_T(j) = \Sigma_{i=0}^{|Q|-1} H_{T-1}(i) p\left(\theta_{T-1} = i \mid \theta_T = j, \omega_1 x\right) \\ &-\Sigma_{i=0}^{|q|-1} p\left(\theta_{T-1} = i \mid \theta_T = j, \omega_{1,T}\right) \log p\left(\theta_{T-1} = i \mid \theta_T = j, \omega_L, x\right) \\ &c_T(j) = T(j) \end{aligned} \tag{18}$$

And calculate the following table as proposed in exercise 22.

	–	**a**	b	a	a	a	a
H_0	0.0						
c_0	1.0						
H_1	0.0	0.0					
c_1	0.0	0.2					
H_2	0.0	0.0	0.0	0.4396	0.4396	0.4396	0.4396
c_2	0.0	0.8	0.16	0.76	0.6	0.6	0.6
H_3	0.0		0.0				
c_3	0.0		0.86				
H_1	0.0			0.65	0.4396	0.4396	0.4396
c_4	0.0			0.26	0.4	0.4	0.4

(19)

4 Exercise

In order to solve this questions we need to iterate with the ISS algorithm.

$$\lambda_1 = \lambda_1 + \delta_1 \tag{20}$$

$$\delta_1 = \frac{1}{M} \log \frac{\tilde{p}\left(\omega_1, c_0\right)}{\tilde{p}\left(\omega_1\right) p_\lambda\left(c_0 \mid \omega_1\right)} \tag{21}$$

We have M=1 since we only have one feature active at any moment. Let's solve the other parts of the equation.

$$\begin{aligned} \tilde{p}\left(\omega_1, c_0\right) &= \frac{2}{5} \\ \tilde{p}\left(\omega_1\right) &= \frac{2}{5} \end{aligned} \tag{22}$$

We calculate $p_{\lambda_1}(y \mid x)$ from the equations on the slides.

$$\begin{aligned} p(y \mid x) &= \frac{1}{Z(x)} \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x, y)\right) \\ Z(x) &= \sum_y \exp\left(\sum_{i=1}^{k} \lambda_i f_i(x, y)\right) \\ p_\lambda\left(c_0 \mid \omega_1\right) &= \frac{1}{2} \end{aligned} \tag{23}$$

We now solve the increment part

$$\delta_1 = \log \frac{\frac{2}{5}}{\frac{2}{5}\frac{1}{2}} = 0.693 \tag{24}$$

Now for λ_2 we apply the same steps.

$$\lambda_2 = \lambda_2 + \delta_2 \tag{25}$$

$$\delta_2 = \frac{1}{M} \log \frac{\tilde{p}\left(\omega_2, c_1\right)}{\tilde{p}\left(\omega_2\right) p_\lambda\left(c_1 \mid \omega_2\right)} \tag{26}$$

With values.

$$\begin{aligned} \tilde{p}\left(\omega_0, c_1\right) &= \frac{1}{10} \\ \tilde{p}\left(\omega_0\right) &= \frac{3}{10} \\ p_\lambda\left(c_1 \mid \omega_0\right) &= \frac{1}{2} \end{aligned} \tag{27}$$

Gives us:

$$\delta_2 = \log \frac{\frac{1}{10}}{\frac{3}{10}\frac{1}{2}} = -0.916 \tag{28}$$

Finally we have $\lambda_1 = 0.693$ and $\lambda_2 = -0.916$

5 Exercise

For this problem we reproduced the experiment on page 61 but with 1000 training samples. In order to do this we used the *sklearn* library and some minor tweaks to sample from the defined gaussian mixture of two unidimensional distributions. Our experiment was to run the EM on a 3 component gaussian mixture with the problem defined means/covariance/weights and calculate a score on how the resultant three mixture assembles the original two component mixture. Score is defined as follows:

$$s = |\pi_1 - 4| + |\pi_2 - 6| + \pi_3 \tag{29}$$

With this score we calculate the distance from the three component mixture to the original mixture with weights (0.4,0.6). We run experiments with the original sample number(50) and 1000, both of them with regulated and non-regulated versions. We repetead experiments 500 times to eliminate as much variance as we could, here are the resutls.

	50 samples	1000 samples
$\gamma = 0.0$	0.1290	0.0010
$\gamma = 0.1$	0.1256	0.0004
$\gamma = 0.2$	0.1269	0.0001

We can see how regardless of our regularization $\gamma = 0.0$ the higher number of samples helps to learn the original mixture significantly better (lower score). When introducing regularization we also see better scores but not as significant, even reporting worse results for the low sample experiments when $\gamma = 0.2$.

These are the pdf for two experiments with $\gamma = 0.2$ and samples(50,1000) against the original mixture.

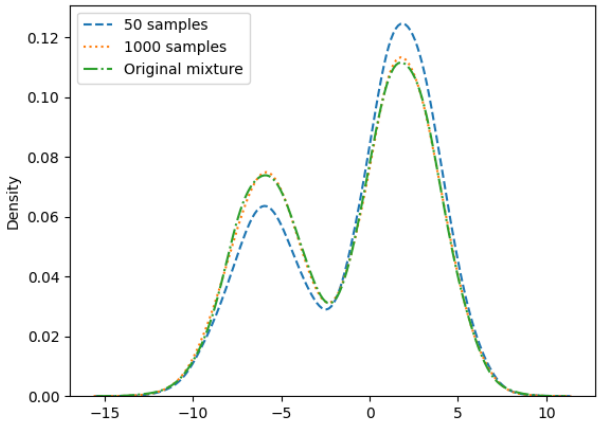

Figure 1: Probability density function

We could say that the 1000 samples experiments correctly learned the original mixture.

6 Practical Exercise

In this exercise we used the SCFG toolkit to learn PCFG on triangle geometry with the Inside-Outside algorithm and the , with Viterbi-Score algorithm. We were ask to continue the exercise statment experiments and iterate on them to improve the results. When running the based commands we got down to 21.63§ % error rate. We got down to 19.83% without needing to add extra data and changing the ratio of negative samples (we created negative samples files for each type with the other types data). This is the baseline experiment confusion matrix.

```
      equi isos righ  Err Err%
equi   860  140    0  140 14.0
isos   410  548   42  452 45.2
righ     0   57  943   57  5.7

Error: 649/3000 = 21.63%
```

Figure 2: Confusion matrix baseline

We saw that the isosceles triangles were the ones with worse results so we did several experiments where we gave a special negative ratio during the isosceles training. With 0.4 ratio for right and equil and 0.1 for isosc we got this results:

```
      equi isos righ  Err Err%
equi 1000    0    0    0  0.0
isos  487  462   51  538 53.8
righ    0   44  956   44  4.4

Error: 582/3000 = 19.40%
```

Figure 3: Confusion matrix

With this combination we didnt improve our isosc triangle results but dramastically improve equil triangles and got to that less than 20% error target. We tried creating more samples with the *genFig* script and more ratios but none of the experiments got better results. We tried adding 1000, 2000, 3000, 4000 extra training samples and ratios from 0 to 0.5 in increments of 0.1.