Machine Learning Statistical Background

Probabilities

Most problems involve stochastic data – 100% prediction may not be possible.

Even when applying machine learning to deterministic problems, may not be able to figure out everything.

Therefore, useful to think about problems in terms of relationships between probability distributions.

Probabilities

Discrete probability distributions:

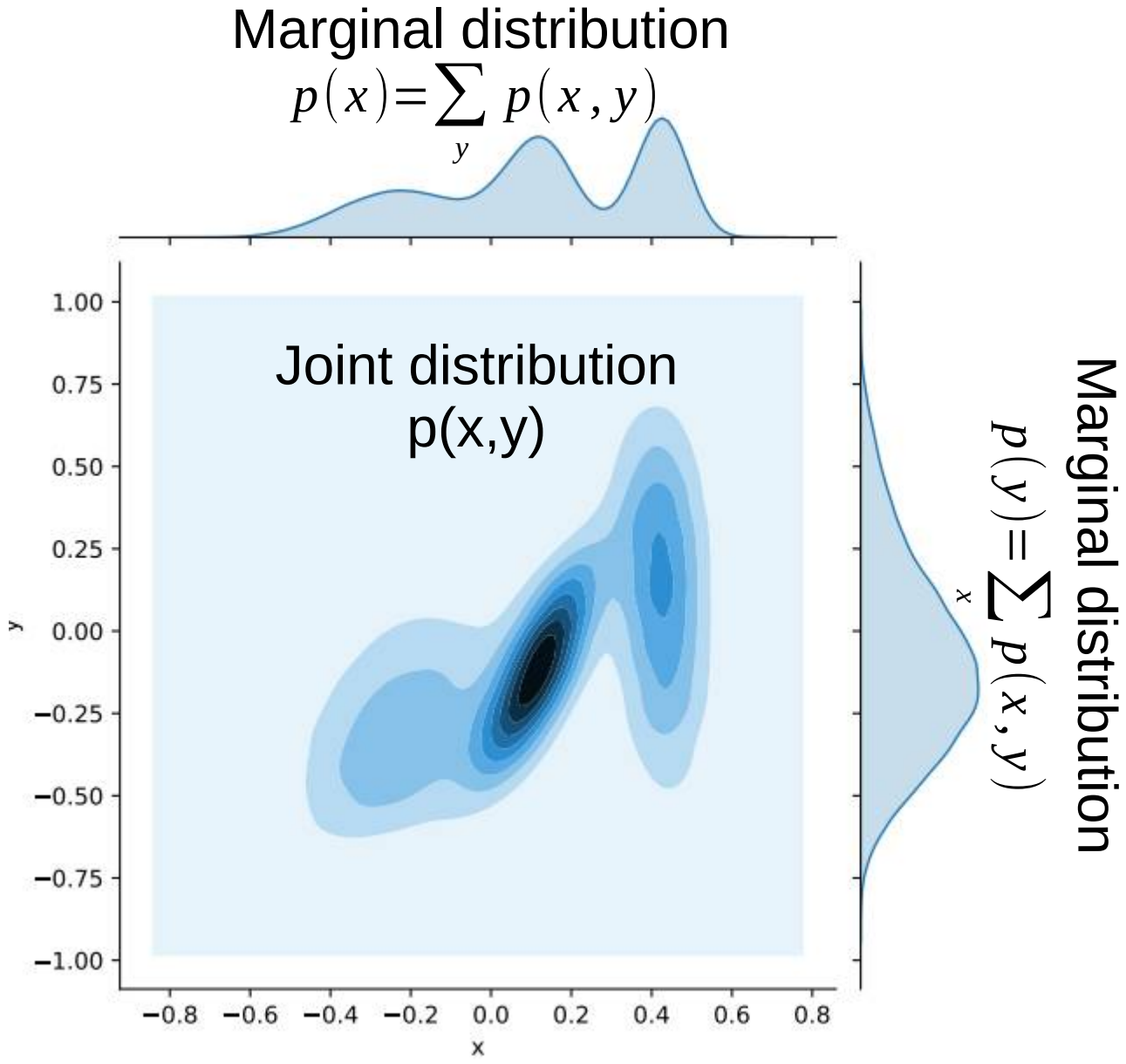
$$\sum_{x,\,y} p(x,y)=1$$

Represent the chance that the variables X and Y take the particular values x,y

Probability **densities**:

$$\iint \rho(x,y)\,dx\,dy=1$$

The integral over a window represents the chance that the variables X and Y take the values within that window.



Probabilities - interpretation

A probability can represent a statement about the **distribution of outcomes** when sampling from a **process**

Flipping a particular coin generates heads with $p_H = 1/2$, and tails with $p_T = 1/2$

A probability can represent a statement about an inference or belief about the state of the world.

If the coin is flipped but I don't see the result, my belief should be $p_H = 1/2$ chance that it is heads and $p_T = 1/2$ chance that it is tails.

After I see the result, my belief will be either $p_H=1$ or $p_T=1$ **for that particular flip**.

Conditional probabilities

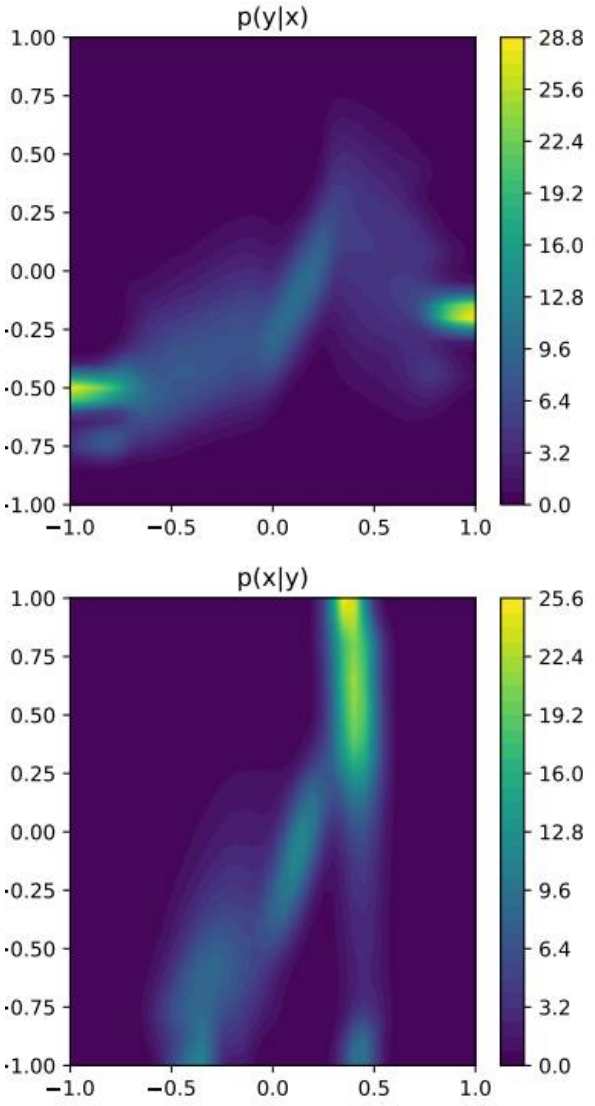
To get ML to evaluate probabilities, frame as an optimization problem. Find function that maps observed **evidence** into optimal distribution of bets **given** the evidence.

Conditional probability: $p(y|x) = \frac{p(x,y)}{p(x)}$

Chance of variable Y taking value y, given observation that variable X takes the value x. Probability of **y** given **x**.

Probabilities reported by an ML algorithm are not **true probabilities of events**

Reflect an **inference**: an optimal belief given evidence and assumptions.



How to specify this optimal inference?

Usual way with a parameterized physical model:

- Simulate model to estimate **p**(**evidence**|**parameters, target**)
- Bayes Rule converts to **p**(**parameters, target**|**evidence**)

$$p(y|x) = \frac{p(x|y)\,p(y)}{p(x)}$$

What if we have no model of the physical process behind the data?

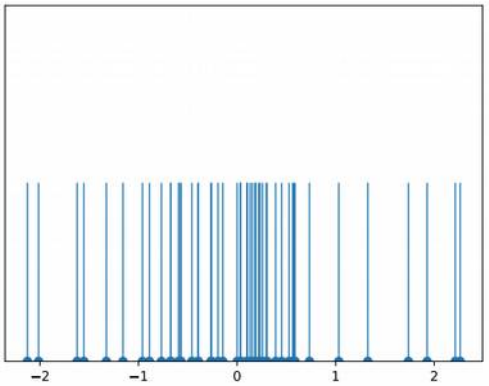
(Non-Bayesian) ML approach: make a general mathematical function taking **evidence** directly to **q**(**target**|**evidence**), where we interpret the output as if it were a probability distribution.

Then try to match **q** with the empirically observed distribution **p**(**target**|**evidence**) in the data.

How to specify this optimal inference?

The **KL divergence** measures a difference between two probability distributions.

$$KL(p(x)||q(x)) = \sum p(x) \log\left(\frac{p(x)}{q(x)}\right)$$



The **empirical distribution** of the data p(x,y): delta function (spike) at every data point.

So e.g. p(x) is 1/N for each bin where there is a data point, zero elsewhere.

This means we can estimate the KL divergence just by taking a sum over each data point:

$$KL(p(y|x)||q(y|x)) = -\frac{1}{N}\sum_{i}^{N} \log(q(y_i|x_i)) + const$$

This gives us the **categorical cross-entropy** or **'log loss'**.

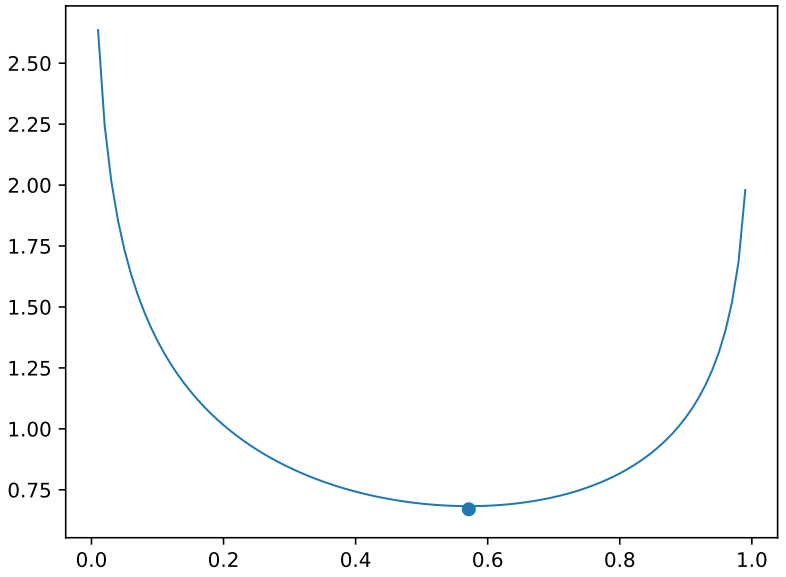
Log-loss example

Dataset of coin flips: [H, H, H, H, T, T, T]

Model is just a constant: $q(H)=1-q(T)=\alpha$

Log loss:

$$L=-\left(\frac{3}{7}\log\left(q(T)\right)+\frac{4}{7}\left(\log q(H)\right)\right)$$



Log-loss interpretation

The log-loss is generally not zero at it's optimal value. Residual is meaningful:

> With best inference possible given the evidence, how uncertain would you still be?

Units are bits ($\log_2$) or nats ($\log_e$), measures the **entropy (H)** of the inferred q(target|evidence).

Roughly, exp(H) is the number of possible targets consistent with the observed evidence (if equally likely).

Log-loss interpretation

What can we do with this?

Lets say we train two models, both predicting **Y**. One gets **X** as input, the other does not.

Residual log losses: H(q(y)) and H(q(y|x))

The difference **bounds** how much x informs us about y:

$$I(y;x) = H(p(y)) - H(p(y|x))$$
$$\geqslant H(q(y)) - H(q(y|x))$$

Mutual information