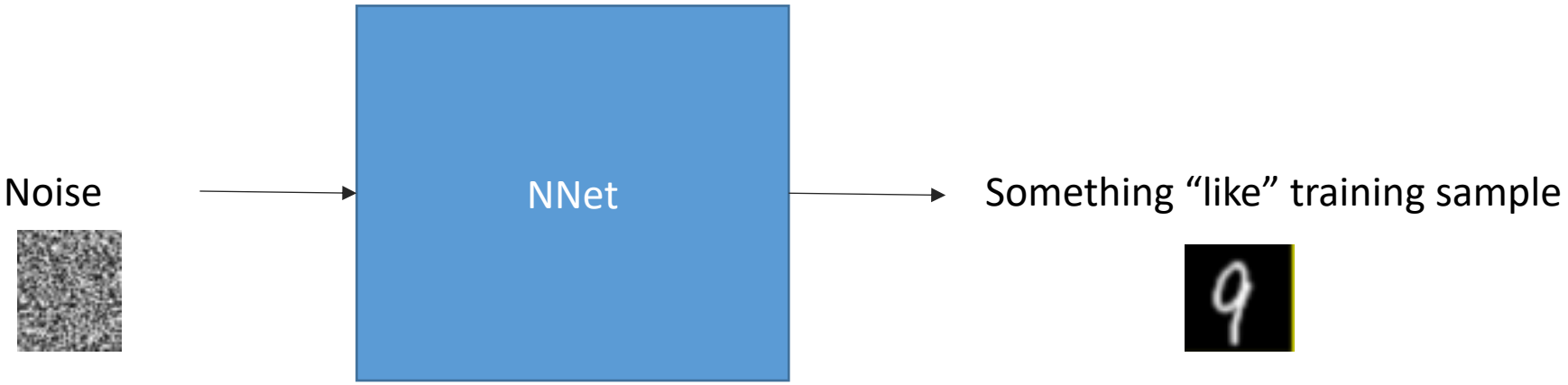
Generative Models, Distribution Distances and GANs

CPSC 452/552

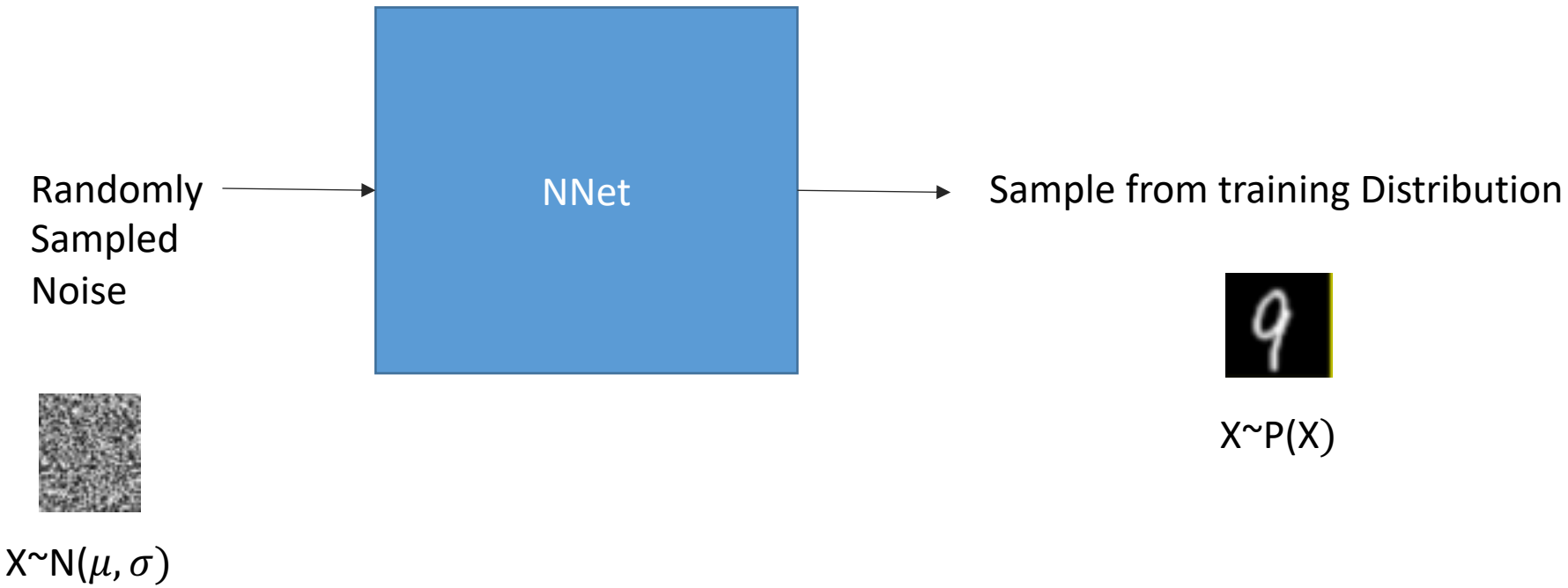
CBB/AMTH 663

Yale

Generative Models

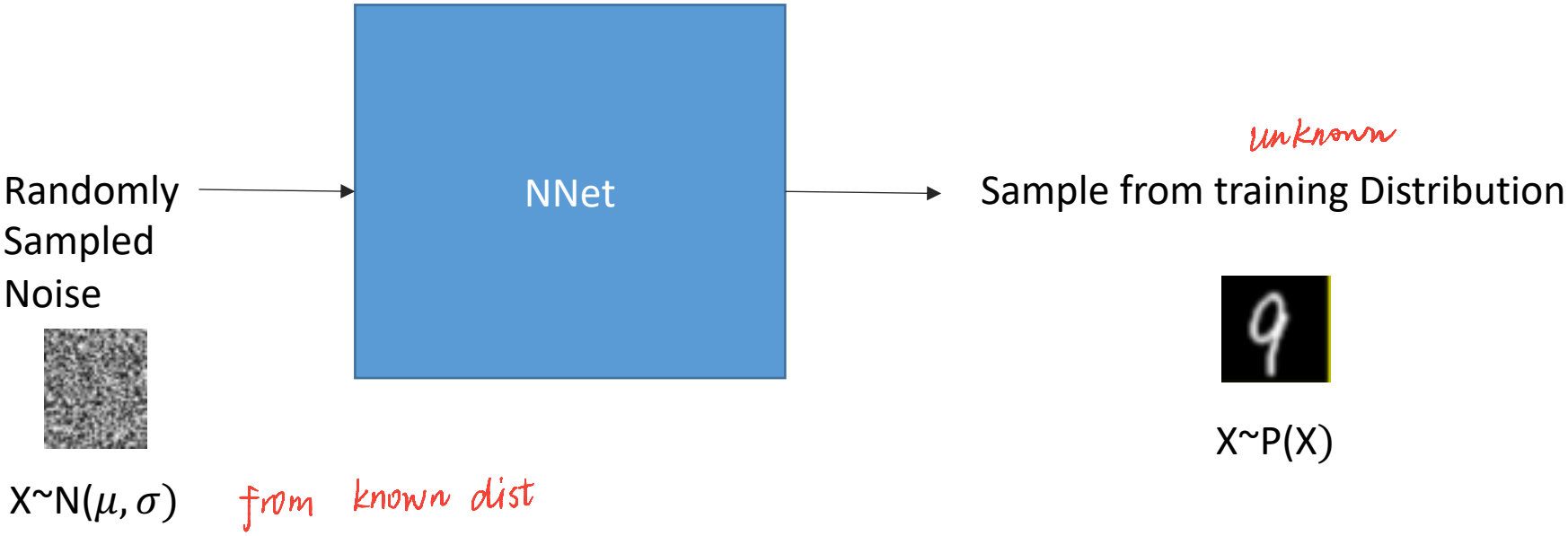


Probabilistic Interpretation



Generative Matching

- Train a neural network with input stochasticity to mimic output distribution

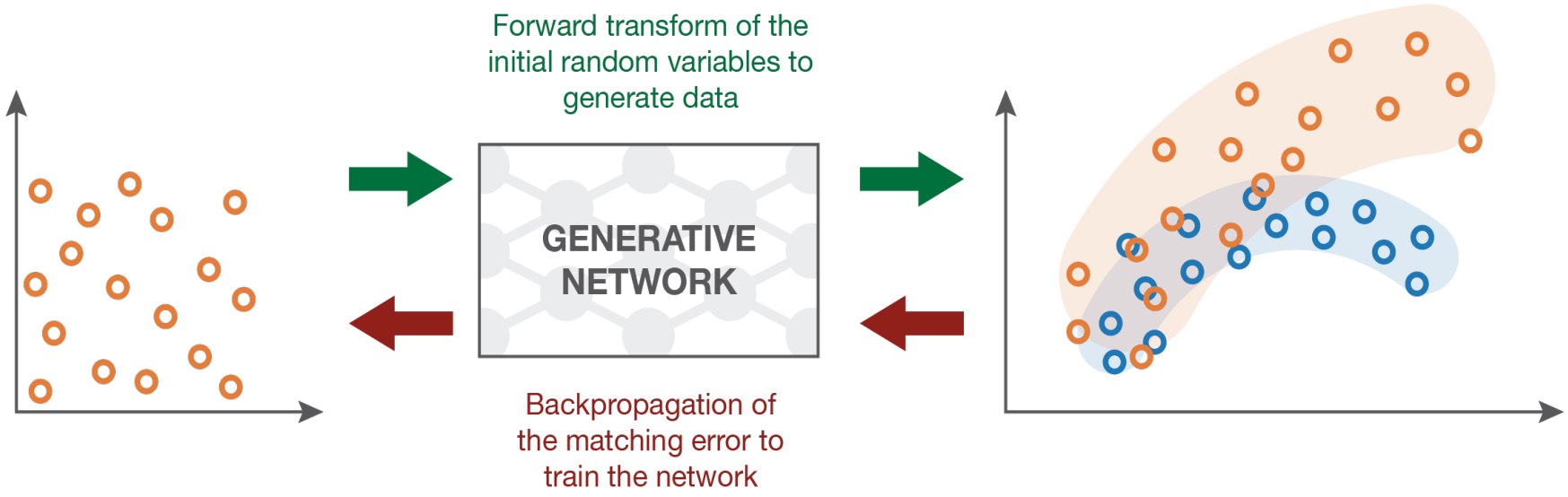


- Penalize by a distribution distance or divergence: KL divergence, MMD distance, wasserstein distance

Outline

1. Distances and divergences between distributions
2. MMD net
3. GANs
4. WGANs

MMD Net



Input random variables (drawn from a uniform).

Generative network to be trained.

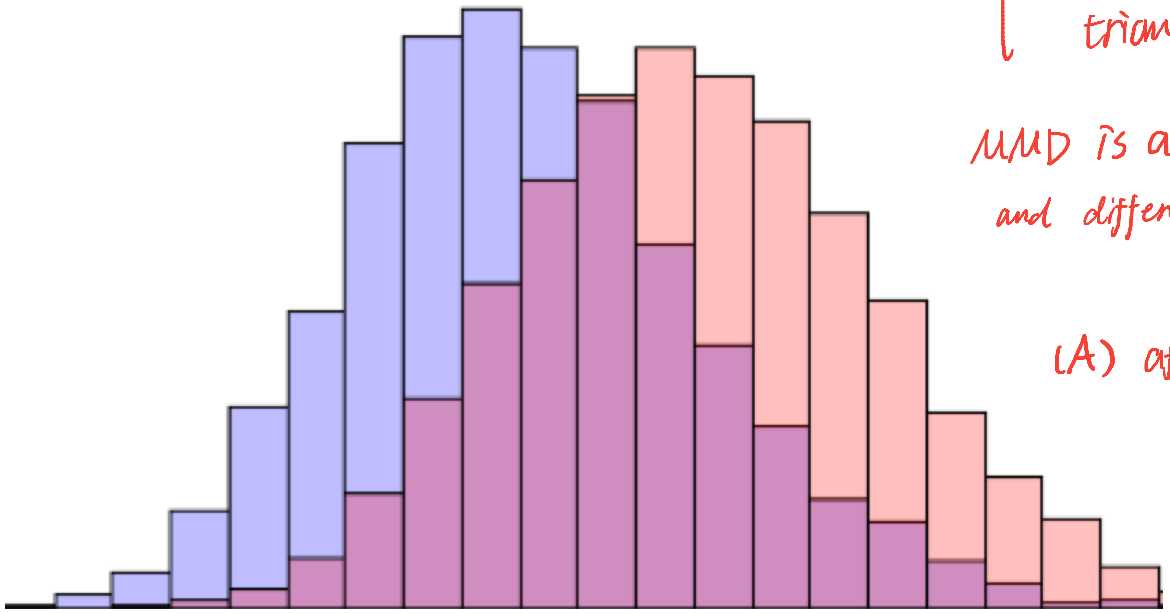
The generated distribution is compared to the true distribution and the "matching error" is backpropagated to train the network.

caculate a MMD distance

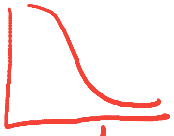
Maximum Mean Discrepancy

distance (metric) { positive semi-definite; symmetric; triangle inequality

MMD is a distance and differentiable



(A) affinity



k is affinity matrix (kernel)

eg. $K(p_i, p_j) = e^{\frac{-\|p_i - p_j\|^2}{2\sigma^2}}$ Gaussian kernel

when $p = q$, MMD = 0

$$MMD(p,q) = \frac{1}{m^2}\Sigma_{i,j\in m}K(p_i,p_j) - \frac{2}{mn}\cdot\Sigma_{i,j}K(p_i,q_j) + \frac{1}{n^2}\Sigma_{i,j\in n}K(q_i,q_j)$$

p is true dist q is generated dist

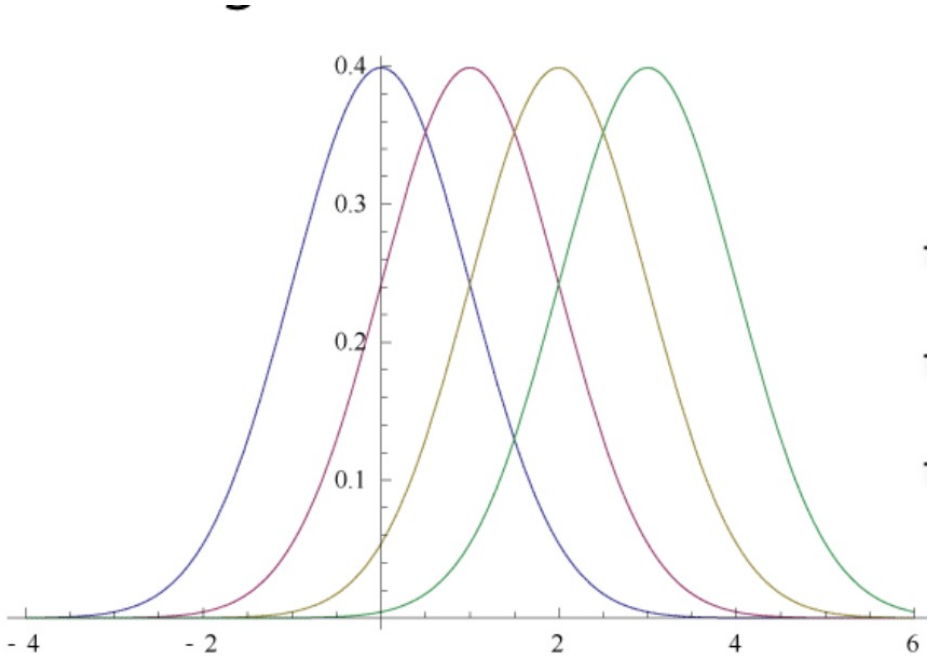
[Dziugaite, Roy, Ghahramani, UAI 2015]

MMD nets

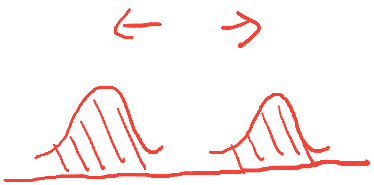
- Train the neural network to transmute noise into desired probability distribution using Maximum Mean Discrepancy optimization. [Dziugaite et al. 2015]
- MMD is a kernel-based 2-sample distribution test for distribution similarity
- **Problem:** This is a batch-level penalty, penalize a whole batch to look like the training distribution

depend on batch size

Can we just measure pointwise L1 distance ?



- Why not?
- What is considered more, areas of high density or low density? (tail)



If 2 dists are disjoint, L1 distance is same comparing 2 dists even far away

Ways of Comparing Probability Distributions

- Cross Entropy
- Divergences
  - KL divergence
  - Jensen-Shannon Divergence

pointwise

contain log damping factor to make tail discrepancy more important

not distance

symmetric

- Distances
  - Earth Mover's Distance (Wasserstein distance)
  - Maximum Mean Discrepancy

Recall: Entropy

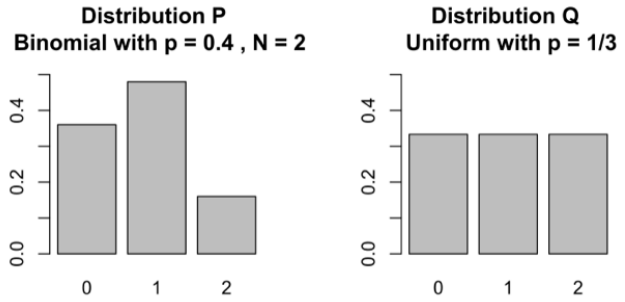
- The expected amount of uncertainty in a distribution
- Given by Shannon entropy

$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

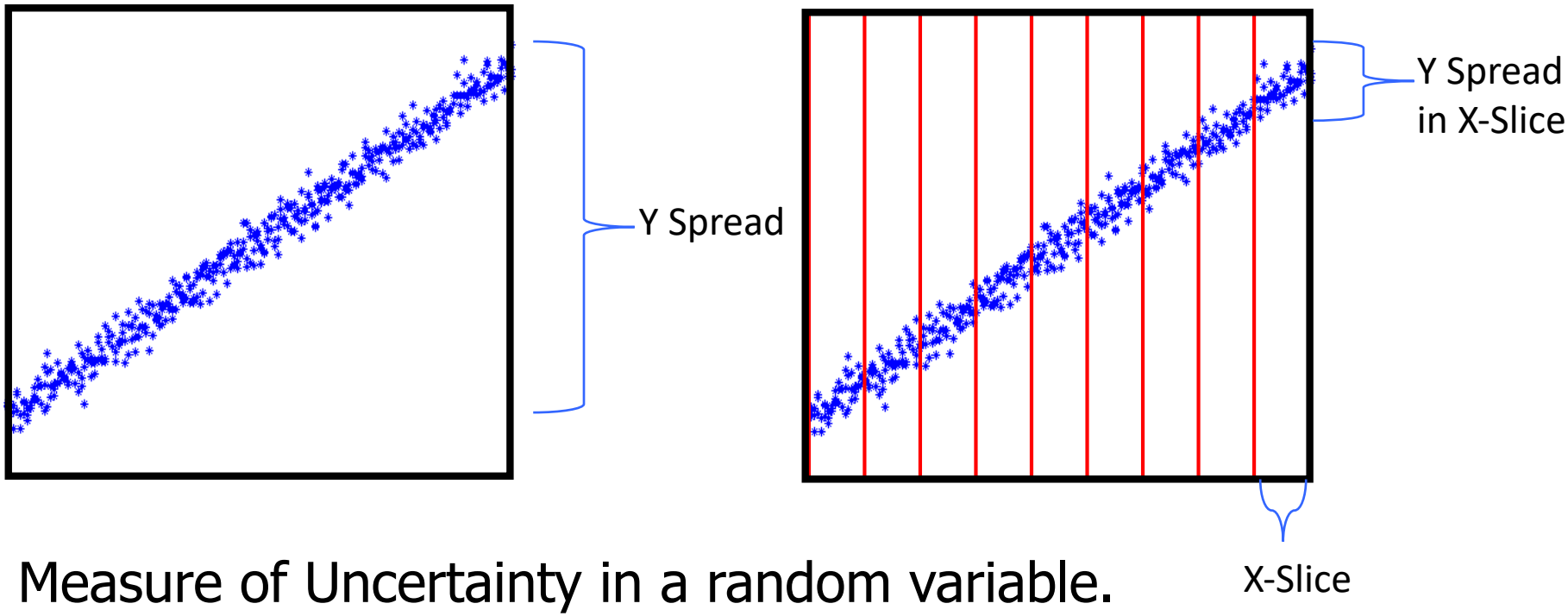
How many bits you need to transmit info in probalistic experiment

usually $b = 2$ base 2

- That is the amount of information you learn by knowing the solution



Entropy as a measure of uncertainty

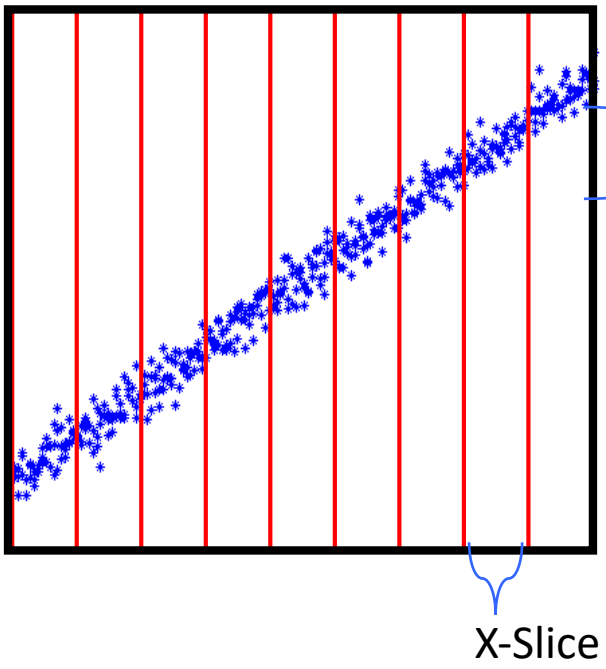


Measure of Uncertainty in a random variable.

$$-\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i),$$

Units of bits tells us how many bits are needed to represent the outcome

Conditional Entropy



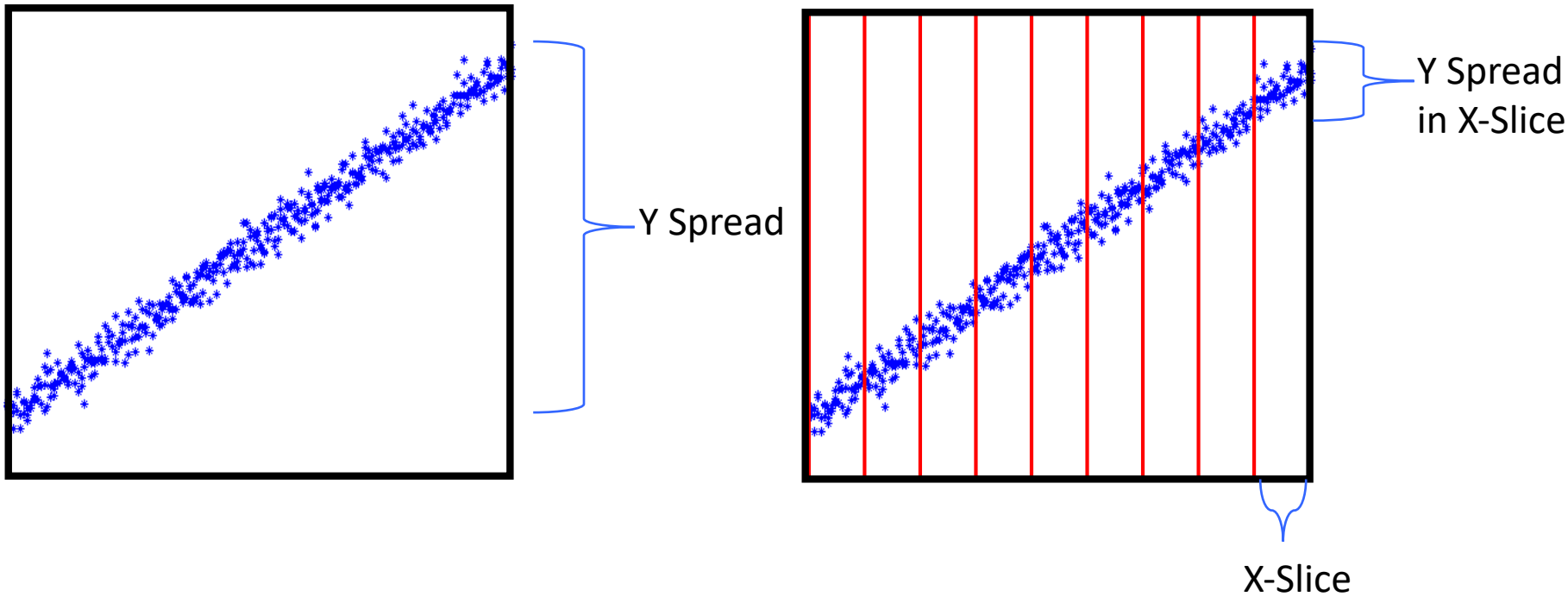
Measure of Uncertainty in a random variable given knowledge about another variable

Can conditioning increase entropy??

No.
H(Y|X)<= H(Y)

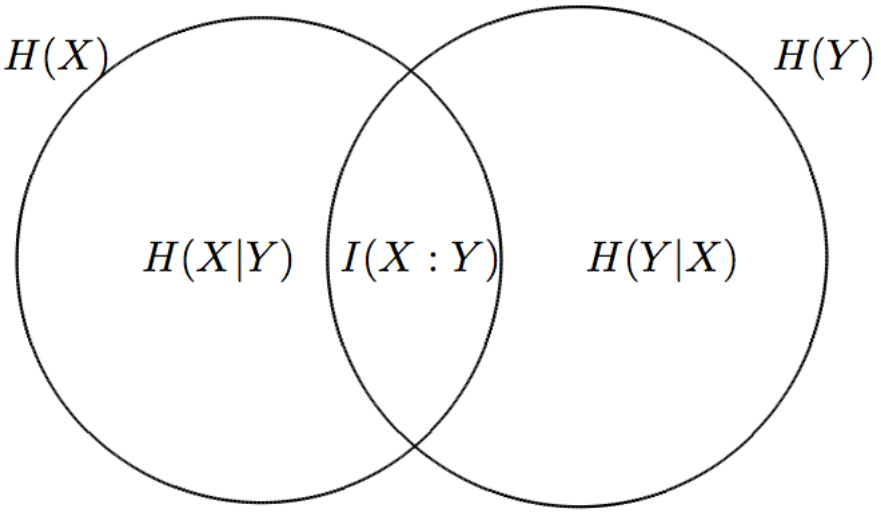
$$
\begin{aligned}
H(Y|X) &\equiv \sum_{x \in \mathcal{X}} p(x)\, H(Y|X=x) \\
&= -\sum_{x \in \mathcal{X}} p(x) \sum_{y \in \mathcal{Y}} p(y|x)\, \log\, p(y|x) \\
&= -\sum_{x \in \mathcal{X}} \sum_{y \in \mathcal{Y}} p(x,y)\, \log\, p(y|x) \\
&= -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log\, p(y|x) \\
&= -\sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log \frac{p(x,y)}{p(x)}. \\
&= \sum_{x \in \mathcal{X}, y \in \mathcal{Y}} p(x,y) \log \frac{p(x)}{p(x,y)}.
\end{aligned}
$$

Mutual Information

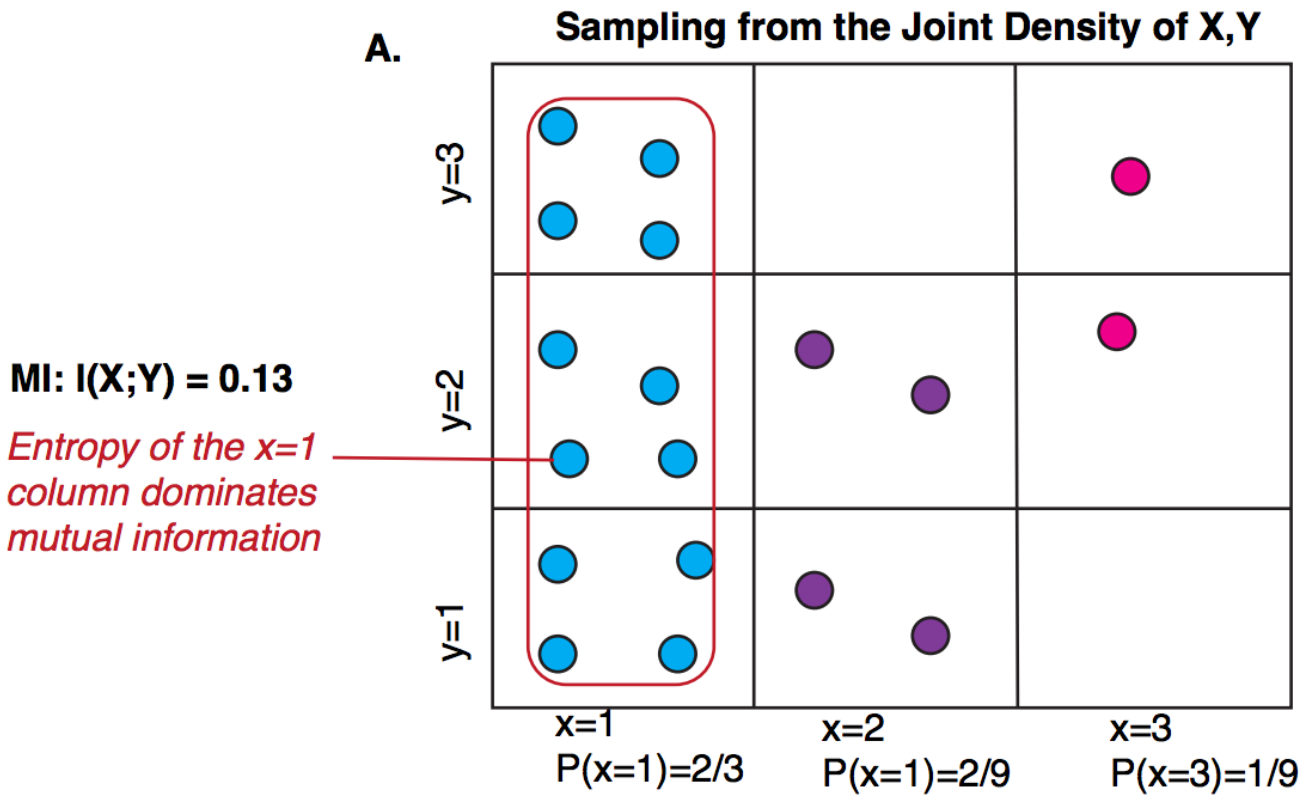


Mutual Information: $I(X,Y) = H(Y) - H(Y|X)$

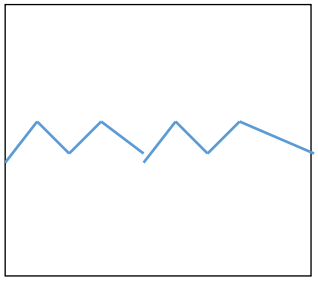
Entropy and Mutual Information



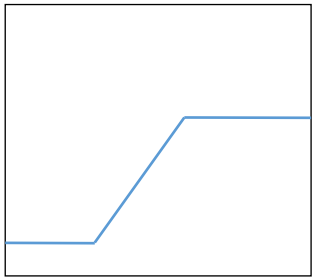
MI Computation

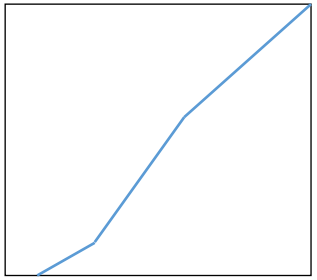


Trends with High and Low MI



Low MI





High MI

Another Definition

- Compares joint to marginal distributions

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

- What is the difference in entropy between joint and product of marginals ?

Cross Entropy

- Given 2 distributions P and Q
- How many bits on average does it take to “encode Q” in a code that is optimized for P

$$H(p, q) = -\sum_{x \in \mathcal{X}} p(x) \log q(x)$$

Encoding a distribution Q

- For instance, if you had 3 outcomes A, B, C,
- q(A)=1/2
- q(B)=1/4
- q(C)=1/4
- You could encode:
  - A as 1 1 bit
  - B as 11 2 bit
  - C as 10 2 bit
- This ensures short messages for communication
- Average number of bits needed is (1/2)*2+(1/2)*1 = 1.5

$(\frac{1}{4}+\frac{1}{4}) \times 2 + (\frac{1}{2}) \times 1 = 1.5$

Suppose you used encoding for Q on distro P

- P is distributed differently than Q
  - p(A)=1/4
  - p(B)=1/4
  - p(C) = 1/2
- Now what is the average number of bits needed?
- Average bits needed = (3/4)*2+(1/4)*1 = 1.5+.25=1.75
- Using Q's code to transmit P takes more bits

$$\left(\frac{1}{4}+\frac{1}{2}\right)\times 2+\left(\frac{1}{4}\times 1\right)=1.75>1.5$$

KL Divergence

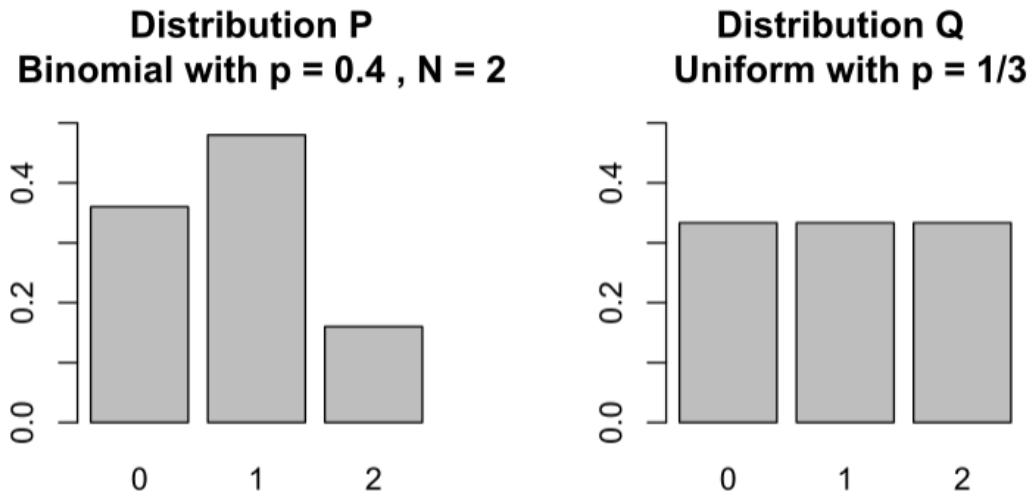
- Expected number of EXTRA bits needed if using samples from P on a code optimized for Q
- Related to cross entropy, also called *relative entropy*

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right) = -\sum_{x \in \mathcal{X}} p(x) \log[Q(x) - p(x)]$$

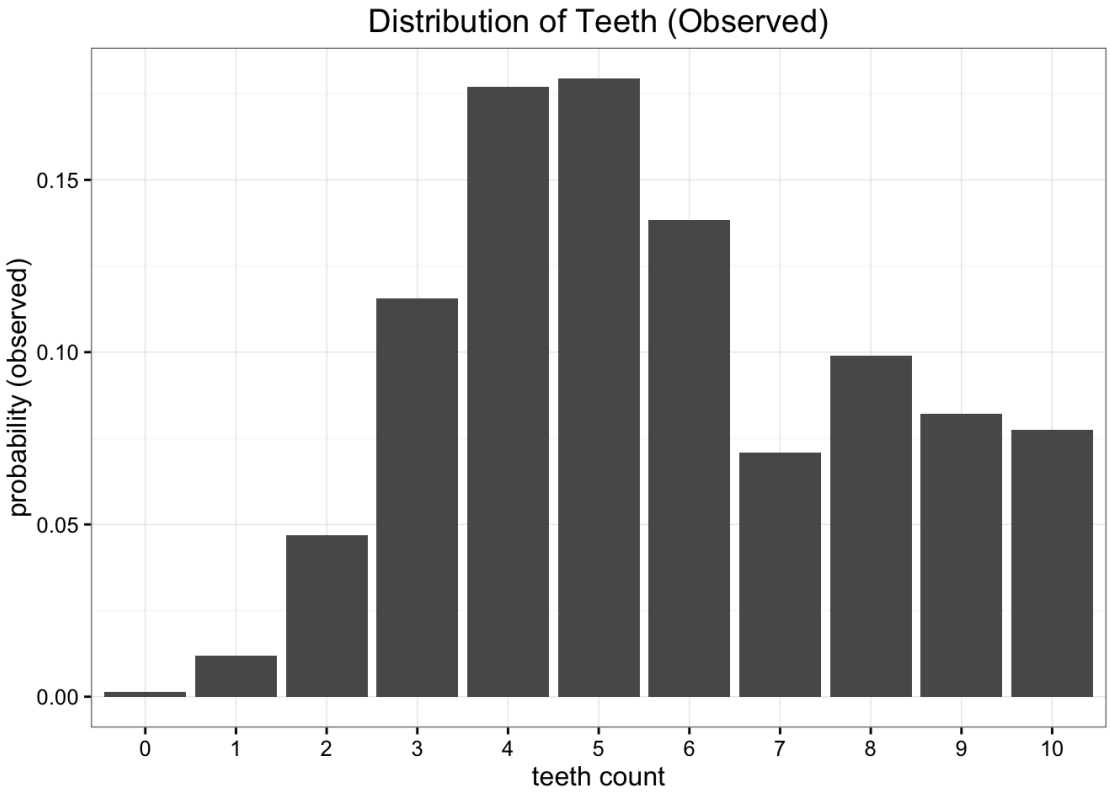
- Note that this is just H(P,Q)-H(P) $= 1.75 - 1.5 = 0.25$

Example

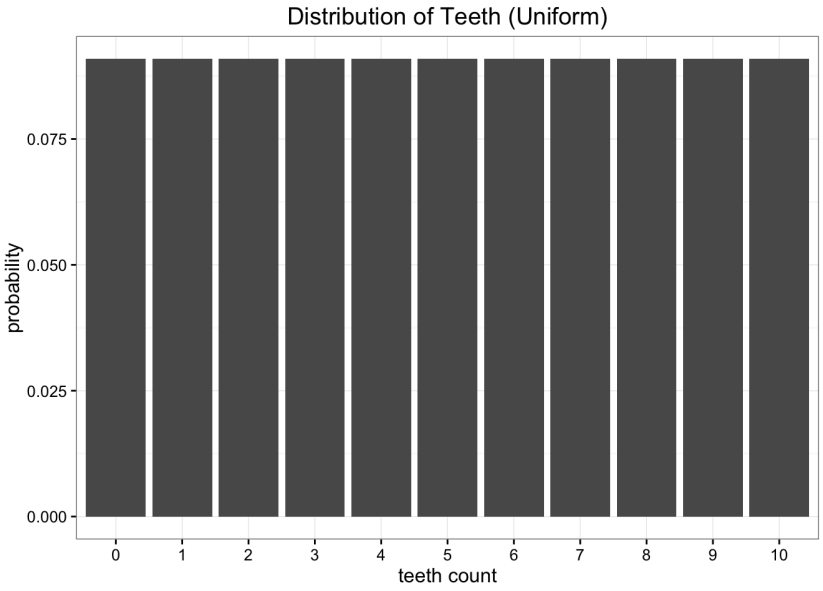
$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

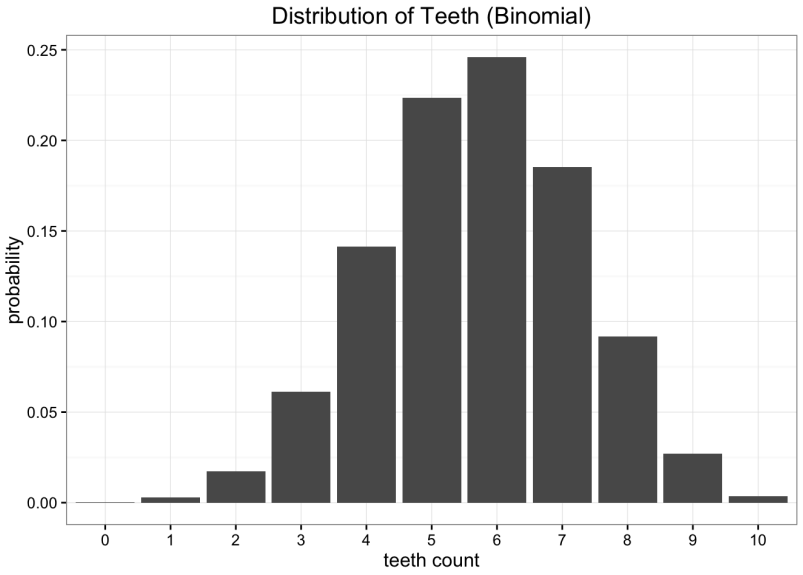


Worm teeth example (from blog)



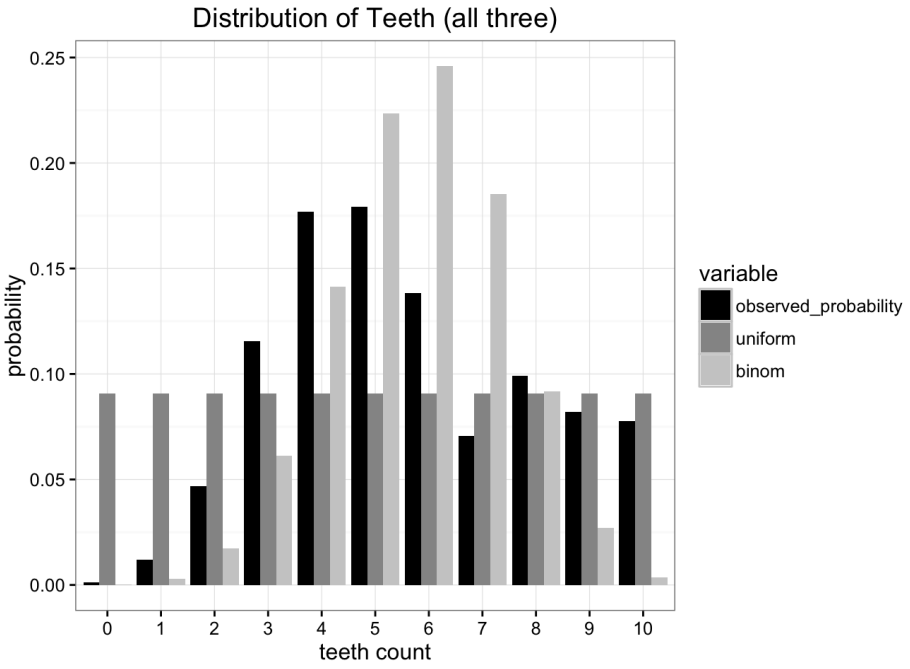
Which simple distribution better approximates this?





Each of these simple distribution has only 1 parameter to convey, N for uniform, and p for binomial

KL divergence of teeth distribution



$$D_{kl}(\text{Observed} \,||\, \text{Uniform}) = 0.338$$

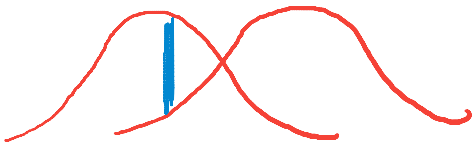
0.338 < 0.477

Uniform is better

$$D_{kl}(\text{Observed} \,||\, \text{Binomial}) = 0.477$$

KL Divergence is not a distance

$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$



more complex form of vertical distance

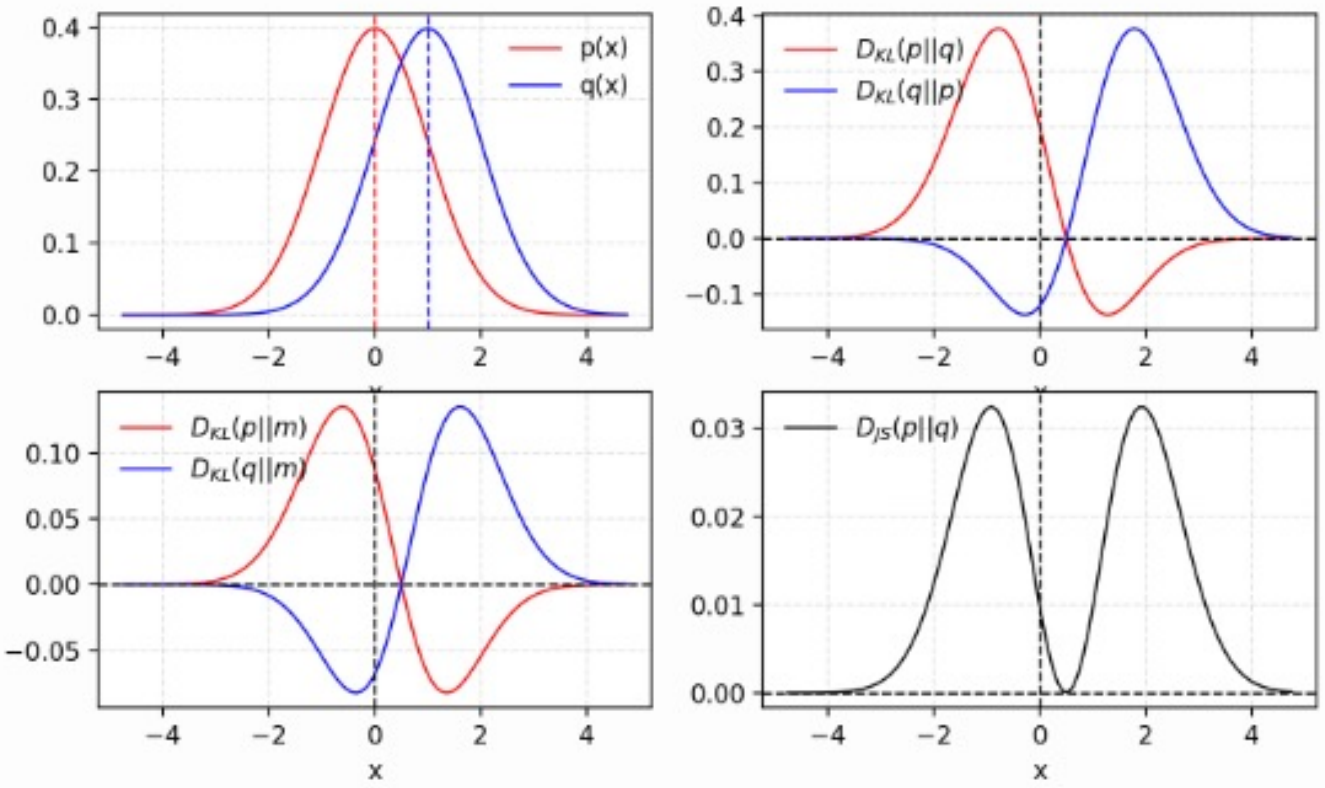
- Can this be a distance?
- Is D(P||Q) same as D(Q||P) NO
- What happens to areas when P is 0? D_{KL} vanish
- What about when Q is 0? D_{KL} infinite
- It also does not follow the triangle inequality

KL Divergence is defined only if $Q(i) = 0$ for any i such that $P(i) > 0$

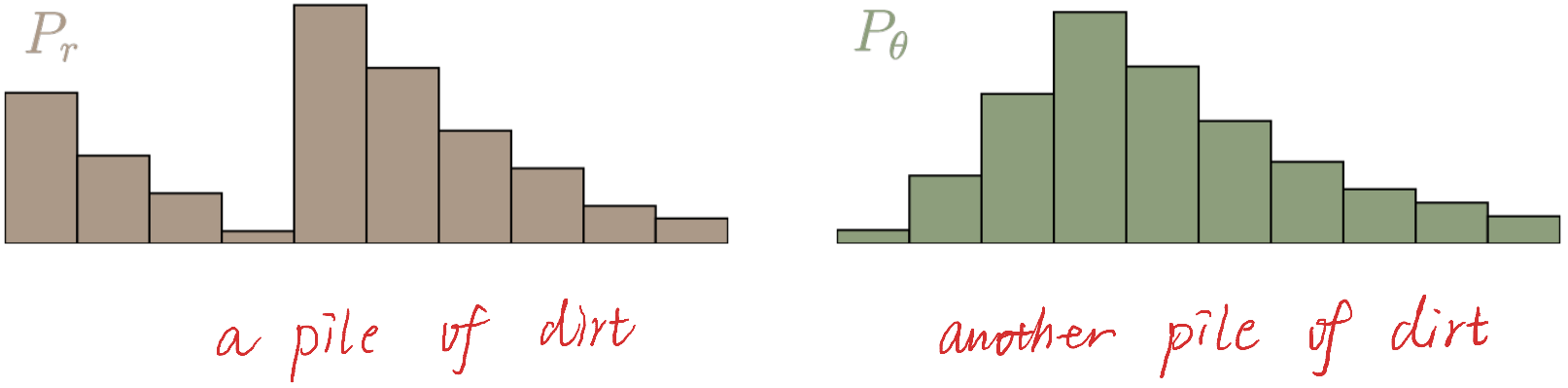
Jensen-Shannon Divergence

symmetric

$$D_{JS}(p\|q) = \frac{1}{2}D_{KL}(p\|\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q\|\frac{p+q}{2})$$

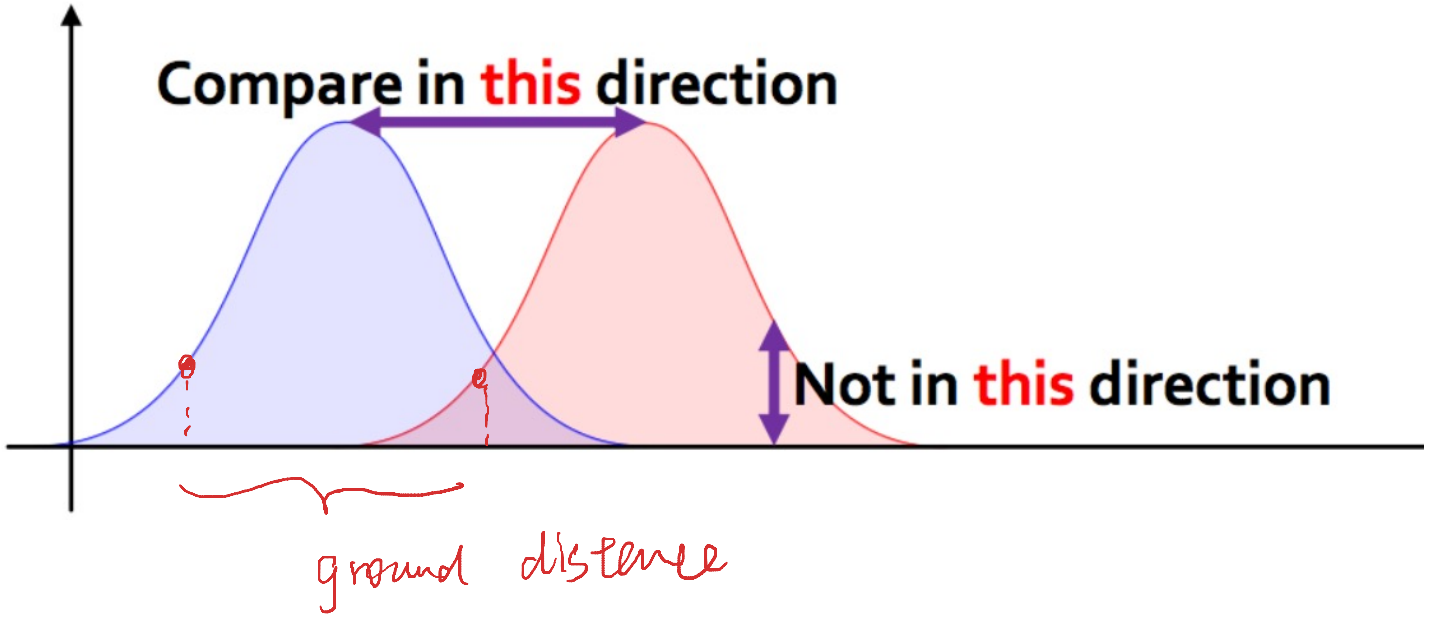


Earth Mover's Distance (EMD)



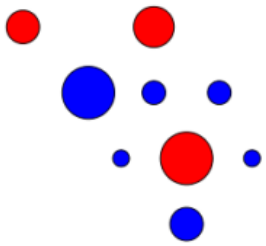
The EMD between two distributions is proportional to the minimum amount of ***work*** *required to convert one distribution into the other.*

Key insight: Ground distance

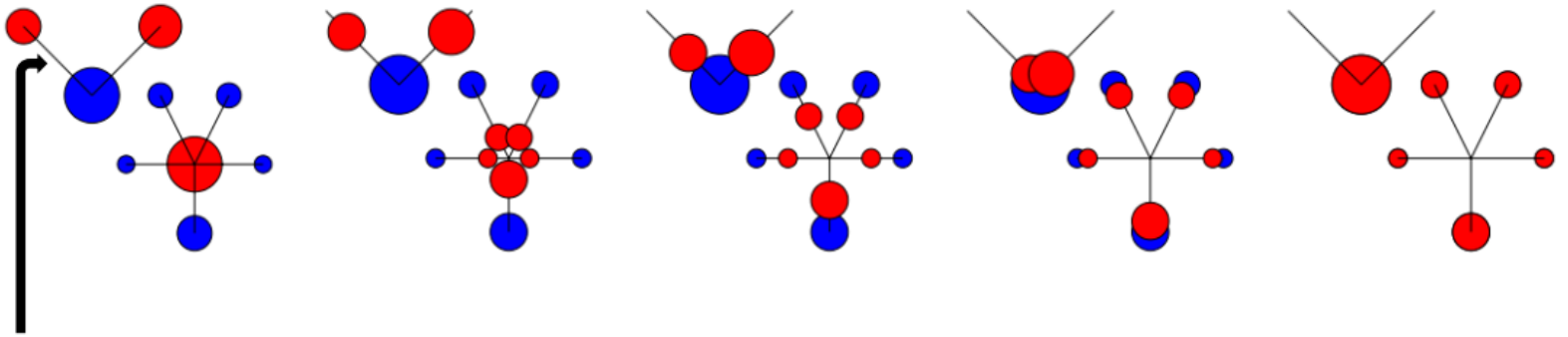


Example

move dirts to fill the holes



- red distribution: “dirt”
- blue distribution: “holes”



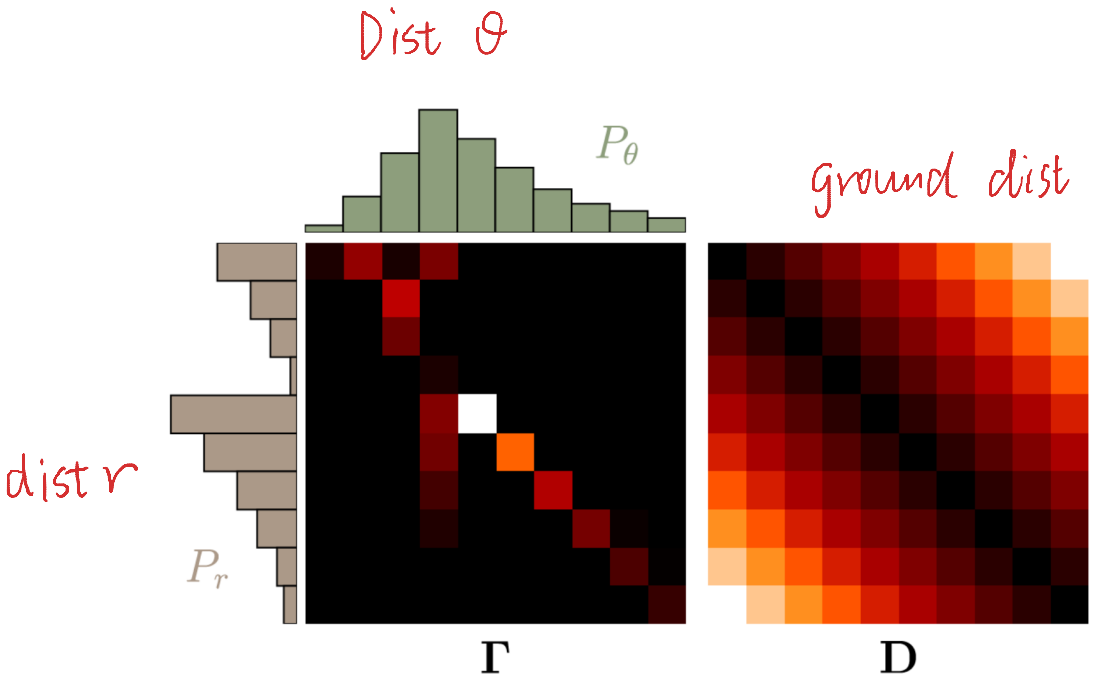
The distance between points (ground distance) can be Euclidean distance, Manhattan…

Valid Transport Plan

cost = $\Gamma \odot D$

objective min Γ

black = 0 cost



constraint

Marginals have to agree:

$$\sum_x \gamma(x, y) = P_r(y)$$

$$\sum_y \gamma(x, y) = P_\theta(x)$$

Object

$$\text{EMD}(P_r, P_\theta) = \inf_{\gamma \in \Pi} \sum_{x,y} \|x - y\| \gamma(x, y) = \inf_{\gamma \in \Pi} \mathbb{E}_{(x,y)\sim\gamma} \|x - y\|$$

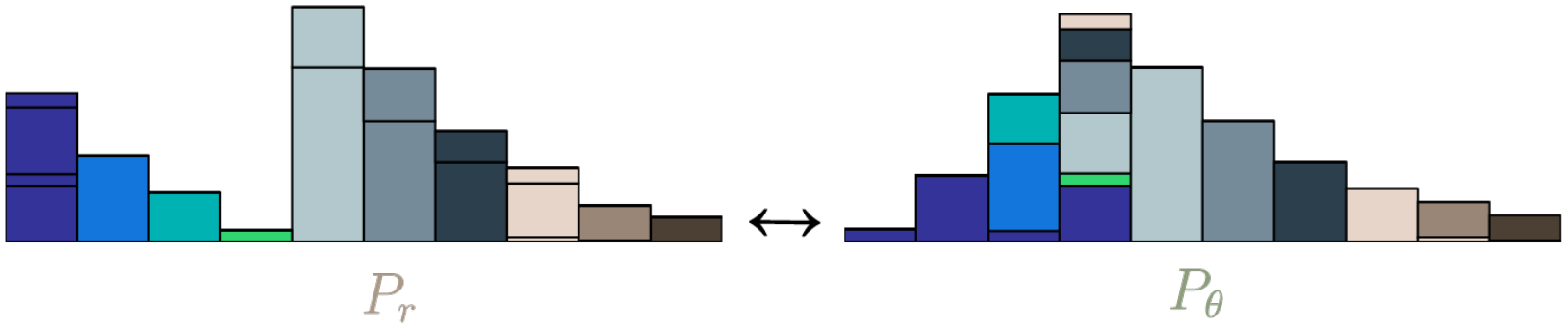
$\gamma(x, y)$ → Γ entry

Linear Programming Solution

- Find a transport plan γ that minimizes $\sum_{x,y} \|x - y\| \gamma(x, y)$
- Subject to the constraint that $\sum_x \gamma(x, y) = P_r(y)$ and $\sum_y \gamma(x, y) = P_\theta(x)$
- This problem has linear constraints and linear objective and therefore can be solved using an LP solver
- However, this is not terribly efficient because the fastest LP solvers are polynomial in the size of the distribution support

problem : not differentiable

Optimal Transport Plan



“Lifting” ground metric to the next level

- Main advantage of EMD is that it “lifts” the ground metric which is defined point-to-point to a set of points
- Can be used to compare *datasets* rather than datapoints
- Images and documents are actually collections of points: pixels and words respectively

Dual of a linear program

dual : can turn a minimization objective to maximization

Minimize

primal form :

$$\begin{aligned} \text{minimize} \quad & z = \mathbf{c}^T\mathbf{x}, \\ \text{so that} \quad & \mathbf{A}\mathbf{x} = \mathbf{b} \\ \text{and} \quad & \mathbf{x} \geq \mathbf{0} \end{aligned}$$

dual form :

$$\begin{aligned} \text{maximize} \quad & \tilde{z} = \mathbf{b}^T\mathbf{y}, \\ \text{so that} \quad & \mathbf{A}^T\mathbf{y} \leq \mathbf{c} \end{aligned}$$

Here c is a vectorized form of D
x is a vectorized form of Γ

$$c = \sum_{(x,y)} \Gamma_{x,y} D_{x,y}$$

A sums up the correct entries of x and ensures its equal to marginal

$$\sum_x \Gamma(x,y) = P_r(y)$$

$$\sum_y \Gamma(x,y) = P_\theta(x)$$

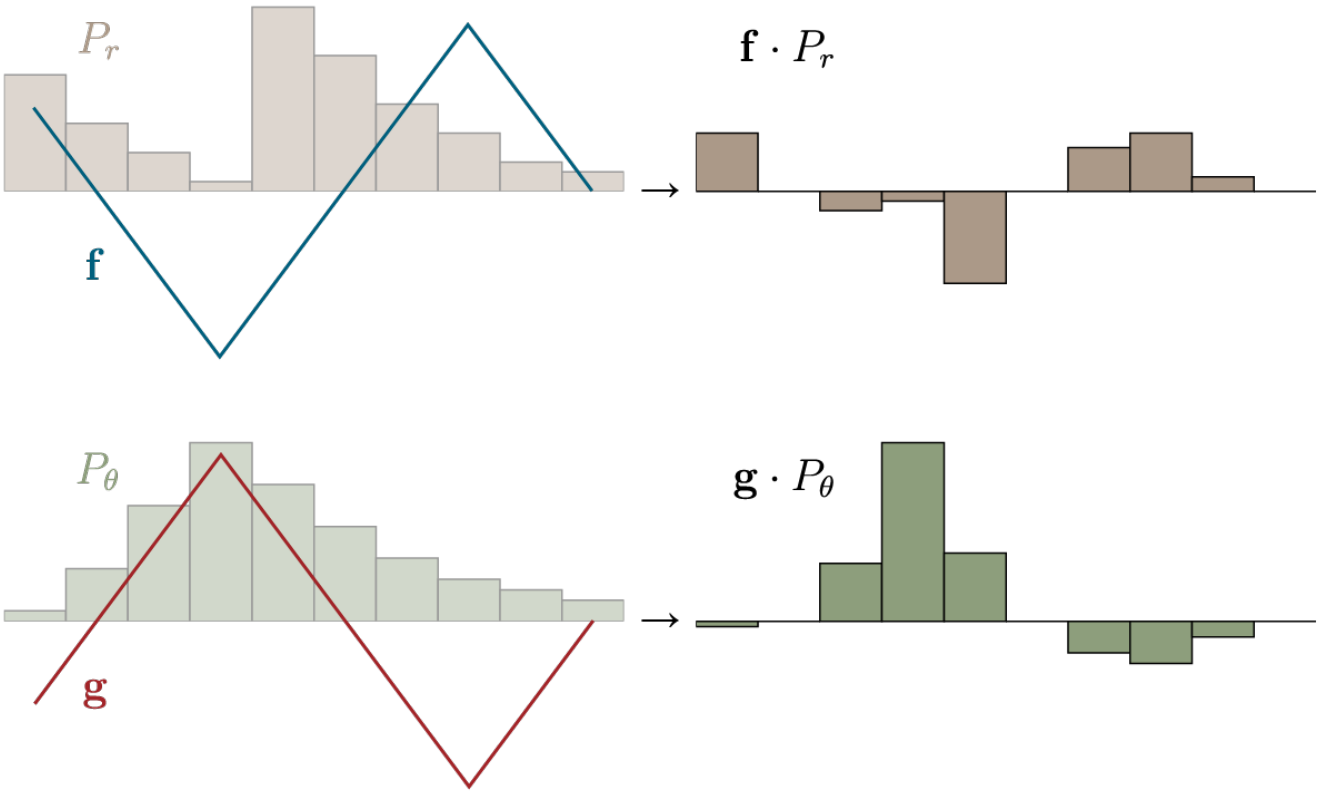
Weak duality:

$$z = \mathbf{c}^T\mathbf{x} \geq \mathbf{y}^T\mathbf{A}\mathbf{x} = \mathbf{y}^T\mathbf{b} = \tilde{z}$$

Strong duality: $z = \tilde{z}$

Dual Form of EMD: Witness Function

take expectation of function f w.r.t dist Pr $E_{x \sim P_r} f(x)$



$$\text{EMD}(P_r, P_\theta) = \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{x \sim P_r} f(x) - \mathbb{E}_{x \sim P_\theta} f(x).$$

f is witness function

Kantorovich Rubenstein Duality

- In the continuous setting, we get a related distance called Wasserstein distance

$$W(p_r, p_\theta) = \inf_{\gamma \in \pi} \iint_{x,y} \|x - y\| \gamma(x, y) \, \mathrm{d}x \, \mathrm{d}y = \inf_{\gamma \in \pi} \mathbb{E}_{x,y \sim \gamma} \left[\|x - y\| \right]$$

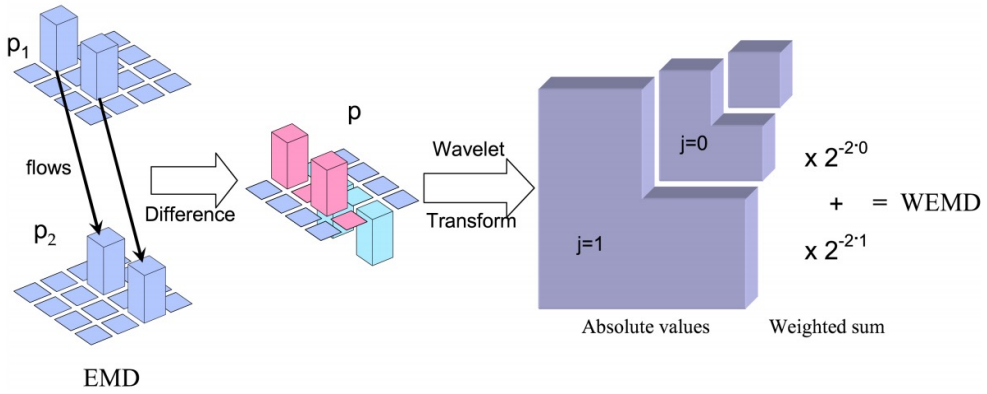
$$= \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{s \sim p_r} [f(s)] - \mathbb{E}_{t \sim p_\theta} [f(t)]$$

This is just a function with the same domain as the probability distributions that has to be maximized!
This version does not need a linear program.

Does not give a transport plan!

Computing EMD with the Dual Form

- Take the difference of two histograms and use a wavelet basis to represent them
- Since wavelets are a rich basis the wavelet transfo



$$d(p)_{wemd} = \sum_{\lambda} 2^{-j(1+n/2)} |p_\lambda|$$

Further reading

- Goodfellow et al., Section 20.10.4
- Lectures on GANs CS 11-785 at CMU
- https://lilianweng.github.io/lil-log/2017/08/20/from-GAN-to-WGAN.html
- https://wiseodd.github.io/techblog/2017/01/26/kl-mle/
- Goodfellow et al 2014 paper
- Arjovsky et al 2017 paper