Information Theory Concepts

outline

- entropy
- conditional-entropy
- mutual information
- KL divergence
- EMD
- MMD

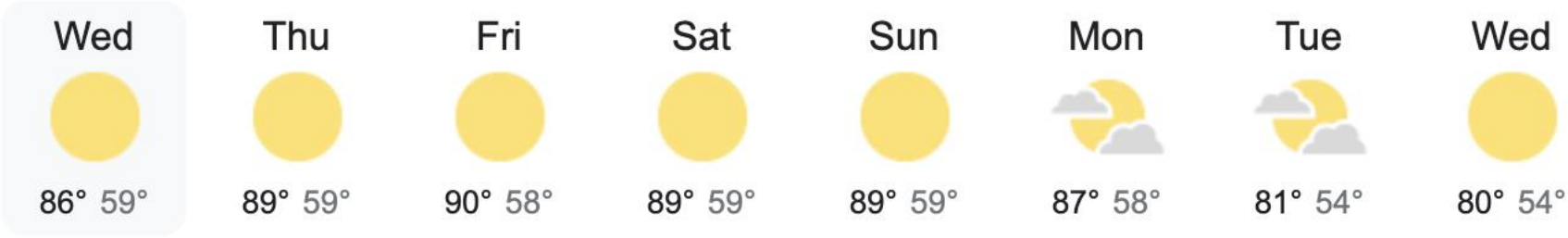
Uncertainty and Information

- What is information?
- One definition is how much do you decrease your uncertainty about an event when you know something?

Is it raining at this minute?

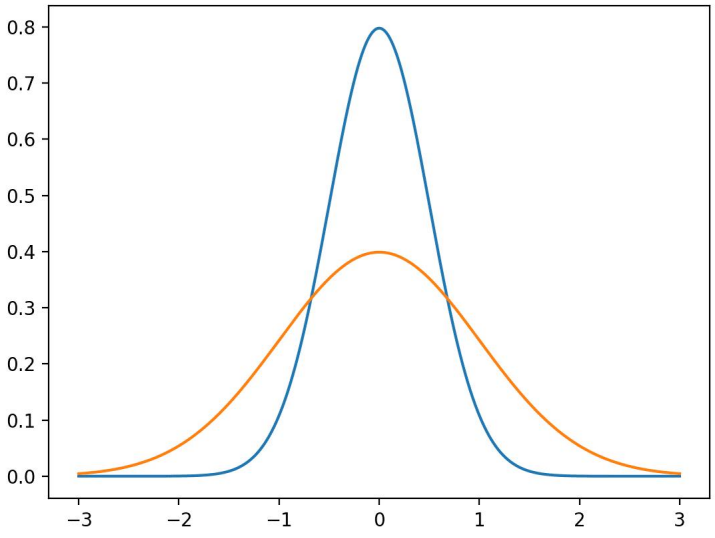
Weekly weather forecast in Seattle

Weekly weather forecast in Phoenix



Quantifying Uncertainty

- What is the amount of uncertainty about the value of a random variable X?



Depends on the probability distribution!

Desired properties of quantification

- Additivity: H(X+Y) = H(X)+H(Y)
  - Two independent events carry twice the surprise
- Positivity uncertainty H(X)>=0
  - No negative surprises ☺
- Certainty: H(1)=0
  - No surprise in certainty
- Permutation invariance

Monotonicity

- A distribution with M uniformly distributed outcomes has less uncertainty than one with M+1

Log(p)

- The only function these constraints!
- Proof
  - $H(p_1p_2)=H(p_1)+H(p_2)$ (additivity)
  - $p_2\ H'(p_1p_2)=H'(p_1)$ (taking the derivative wrt p_1)
  - $H(p_1p_2)+p_1p_2H''(p1,p2)=0$ (derivative wrt p_2)
  - $H'(u)=-uH''(u)$ ($u= p_1p_2$)
- Differential equation leads to $H(u)=k \log U$

Shannon Entropy

- Given by:

$$H(X) = -\sum_{x} p(x) \log_2 p(x)$$

H(X) is expressed in bits and can take any non-negative real numbered value.

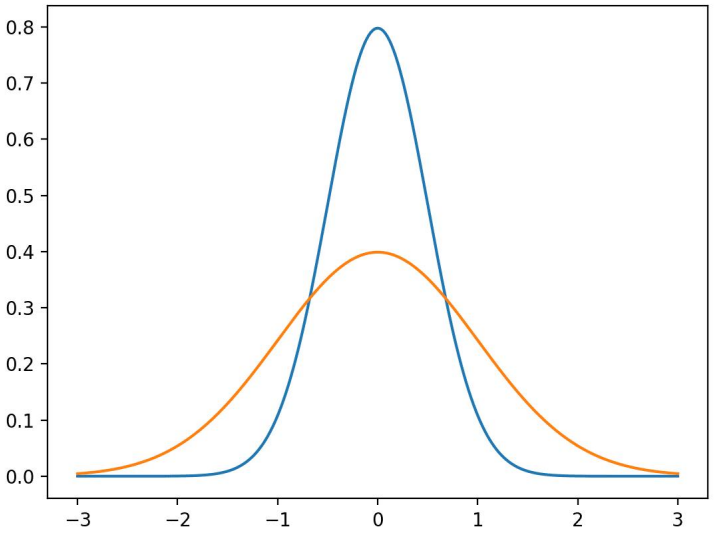
- Ex: Fair coin
- P(heads) = ½ , P(tails)=1/2
- H(coin) = -(1/2) log(1/2) –(1/2)log(1/2)

= $\log_2 2 = 1$

Ex2: Fair dice

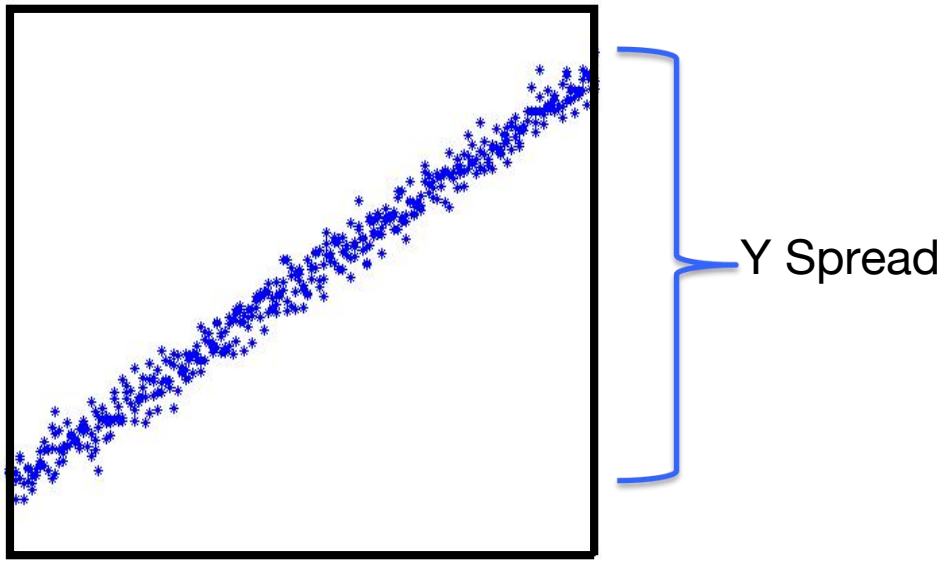
- P(i) = 1/6, i=1, 2, 3, 4, 5, 6
- H(coin) = -(1/6) log(1/6) *6= log(6)=2.6
- More entropy = more surprise!

Entropy of Gaussians



$$\ln\left(\sigma\sqrt{2\pi e}\right)$$

Entropy as spread of a random variable

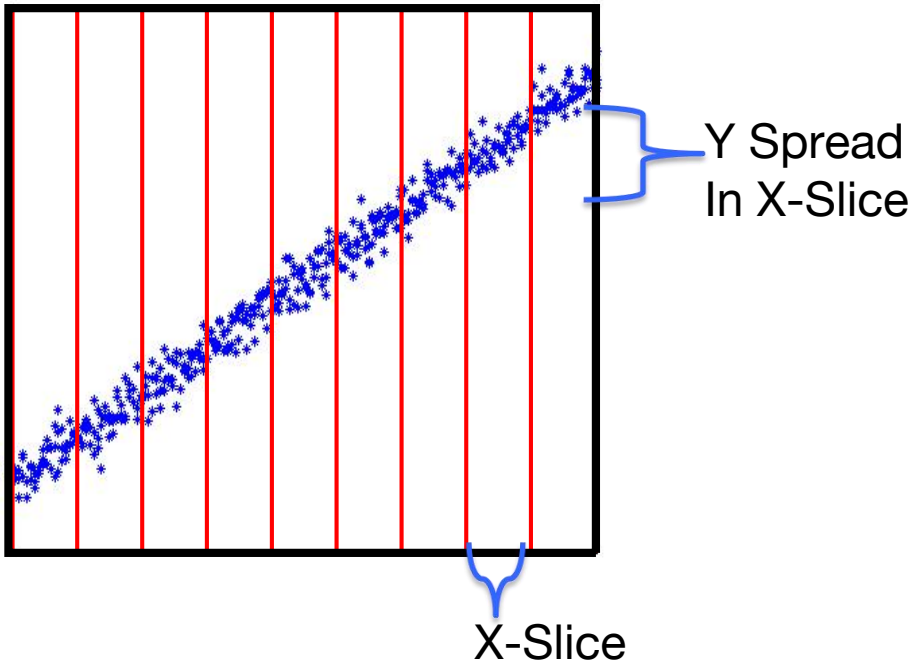


Measure of Uncertainty in a random variable.

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

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

Conditional Entropy

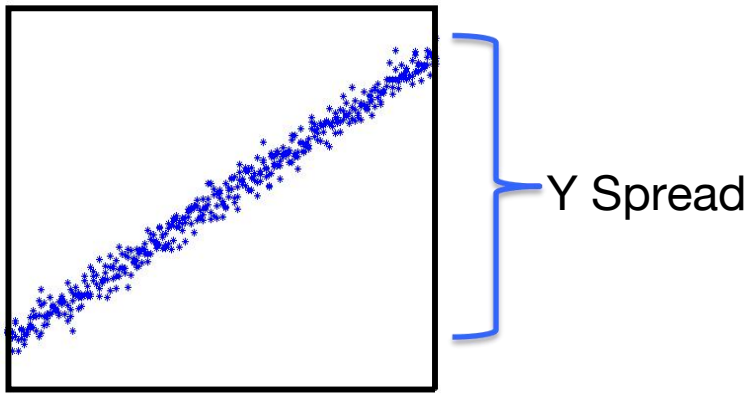


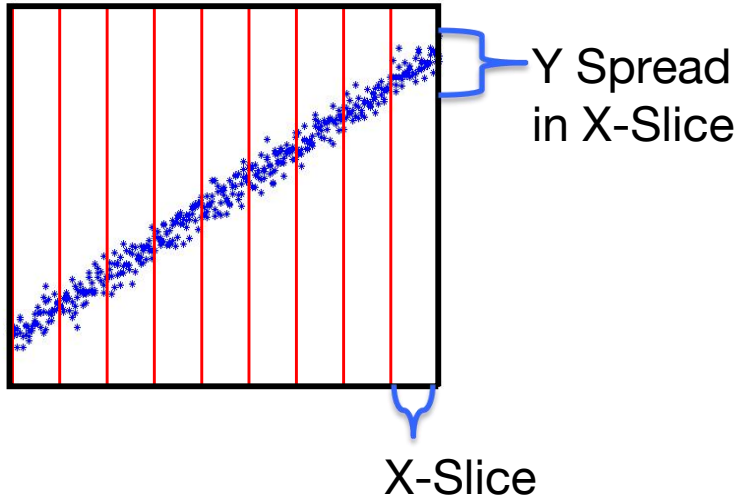
Can conditioning increase entropy??

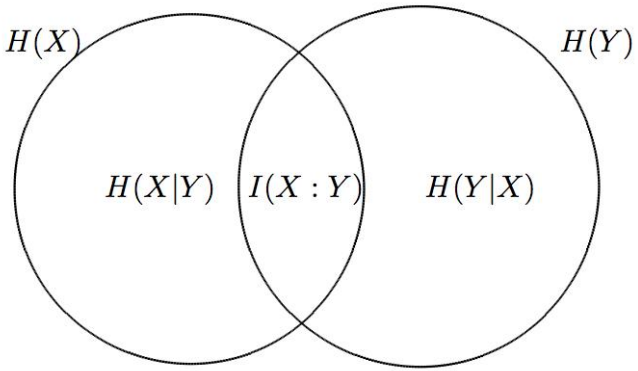
No.
H(Y|X)<= H(Y)

Measure of average uncertainty in a random variable given knowledge about another variable

Mutual Information

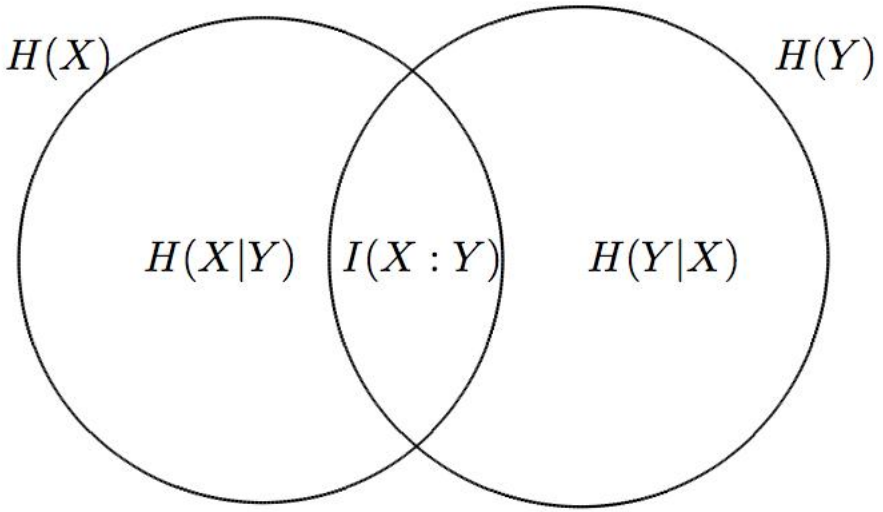




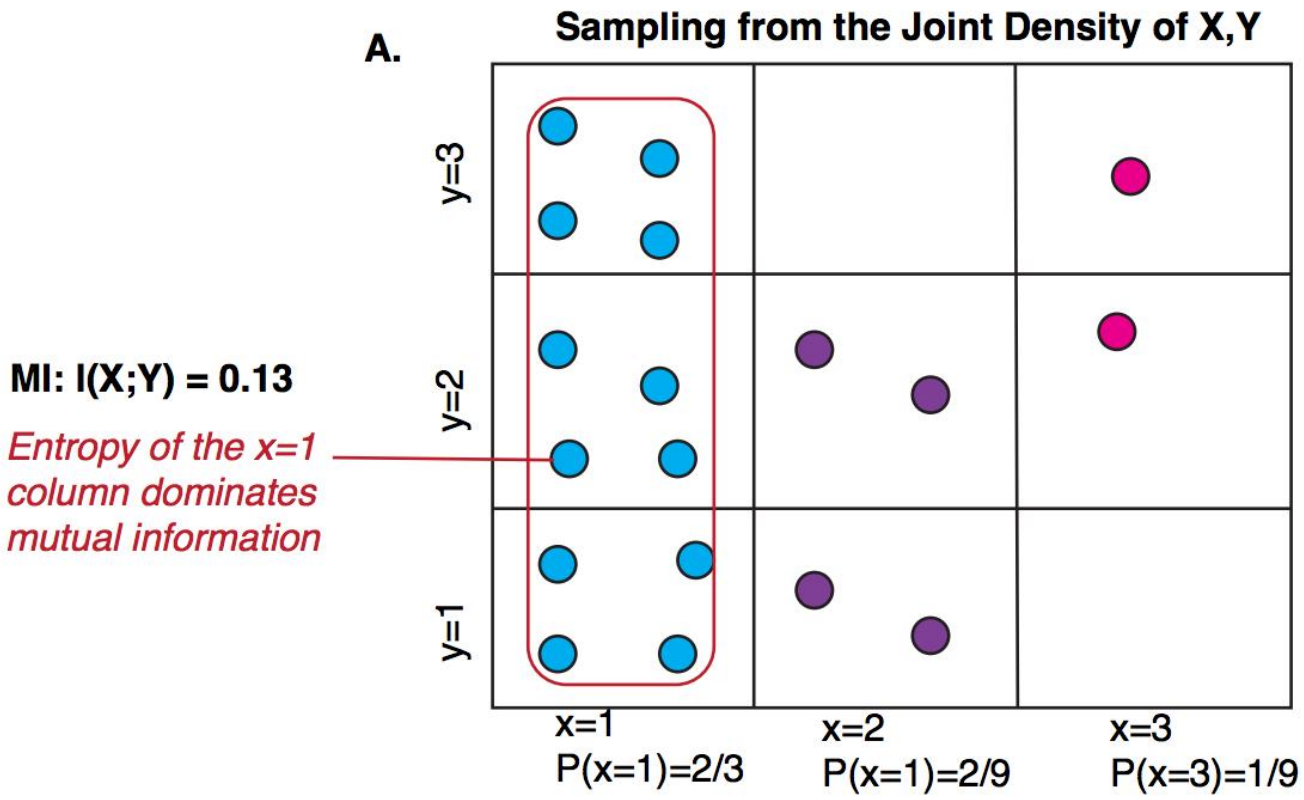


Mutual Information: I(X,Y) = H(Y) –H(Y|X)

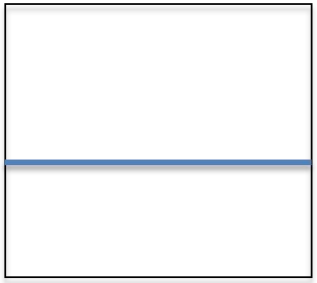
Entropy and Mutual Information

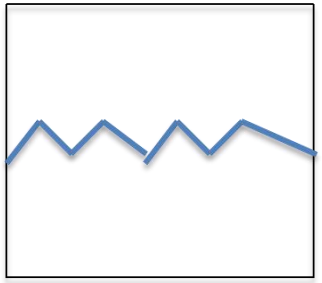


MI Computation

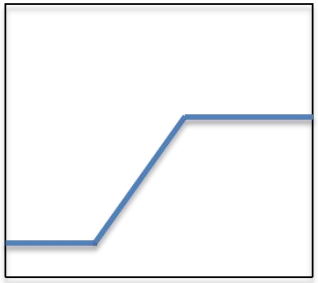


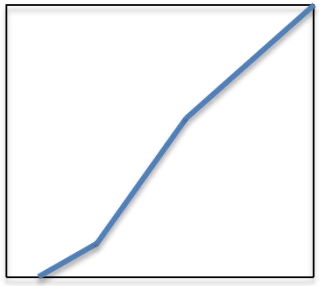
Trends with High and Low MI





Low MI





High MI

Another Definition

- Compares joint to marginal distributions

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

Properties of Mutual Information

- I(X;Y)>=0
- Data processing inequality:

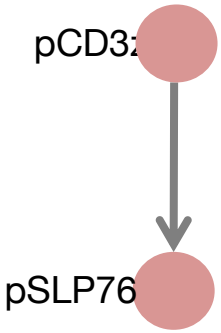
$$X \rightarrow \boxed{\text{Channel}} \rightarrow Y \rightarrow \boxed{\text{Processing}} \rightarrow Z$$

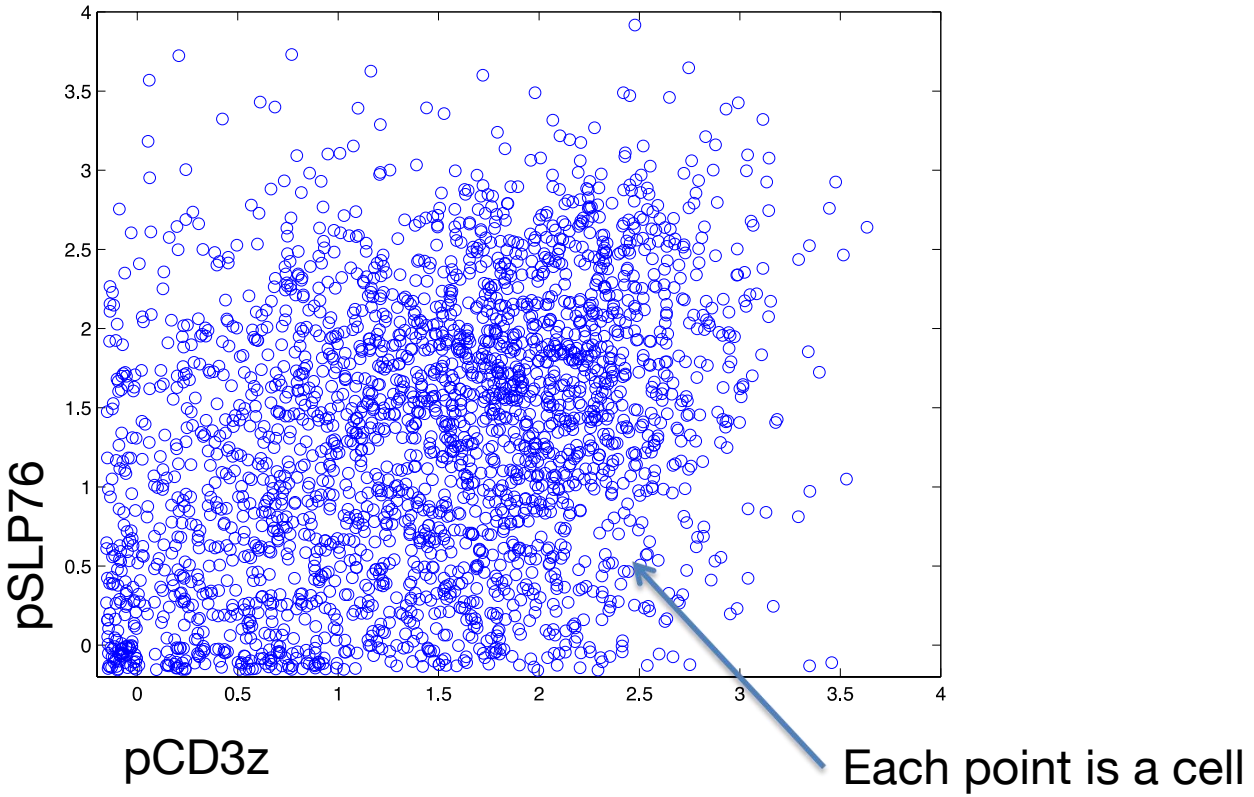
- I(X;Z)<=I(Y;Z)
- Conditioning on a third variable can increase or decrease MI between two variables

Exercise

- Write a function that computes entropy of a variable
- Use it to compute mutual information

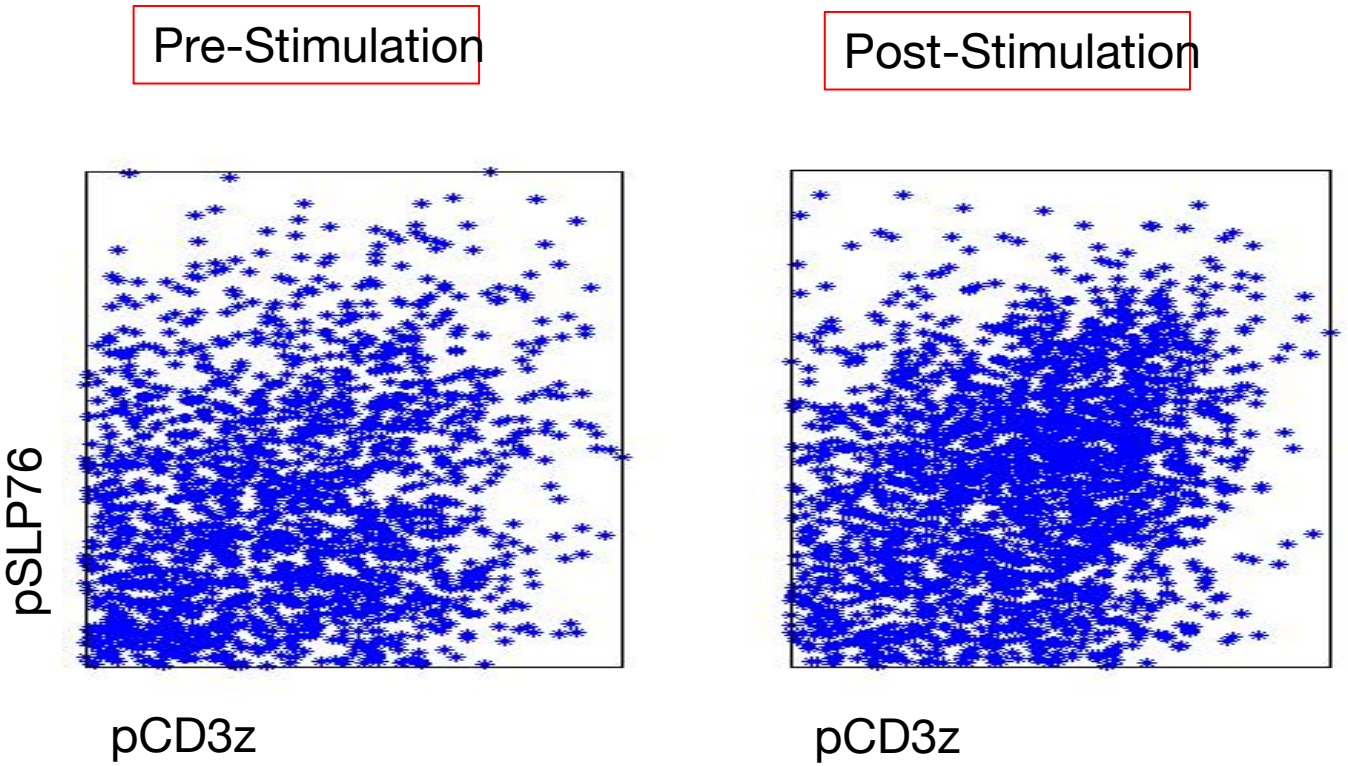
Data distribution





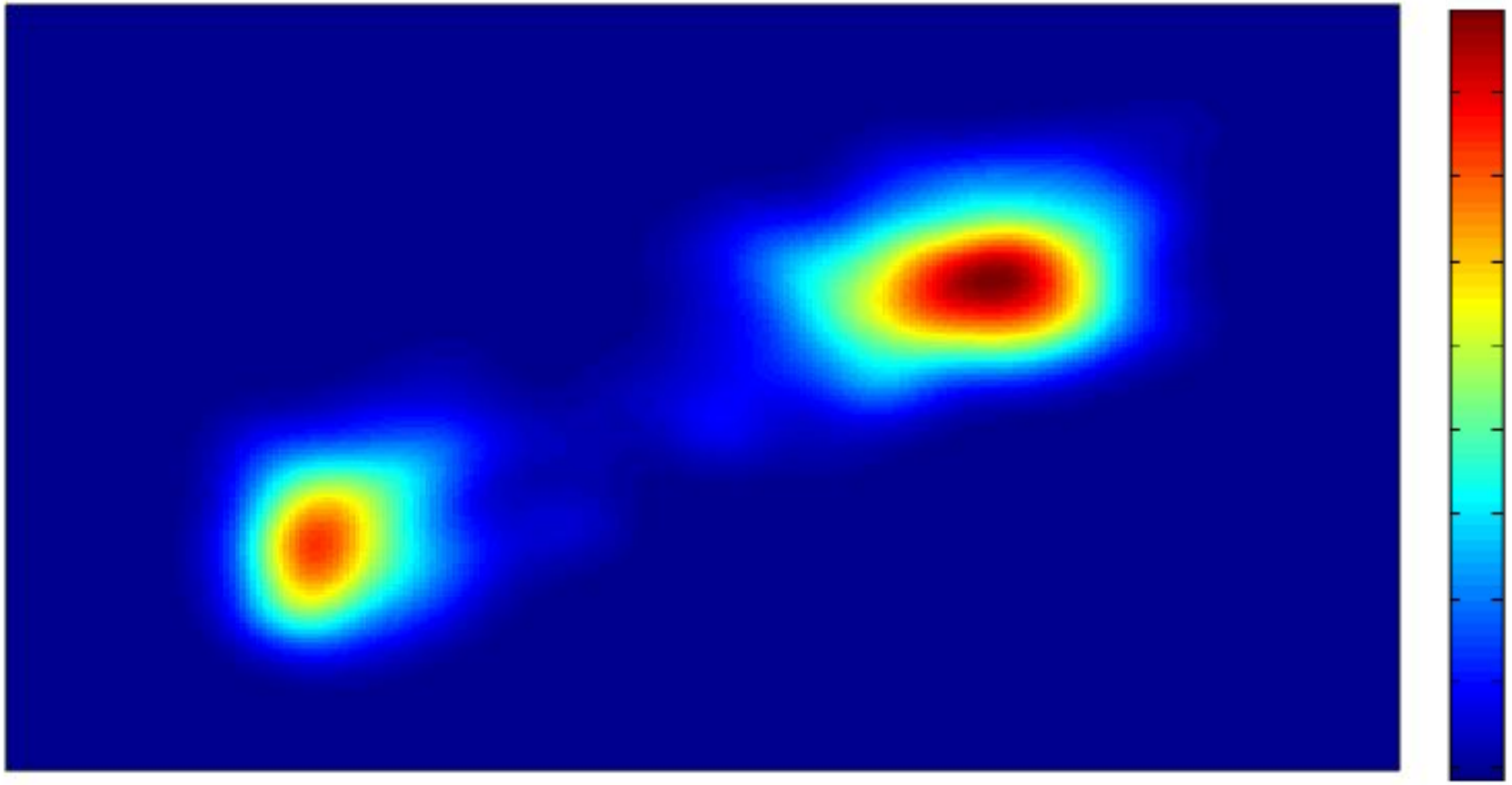
Units of measurement: log-scale transformed molecule counts

Data Density



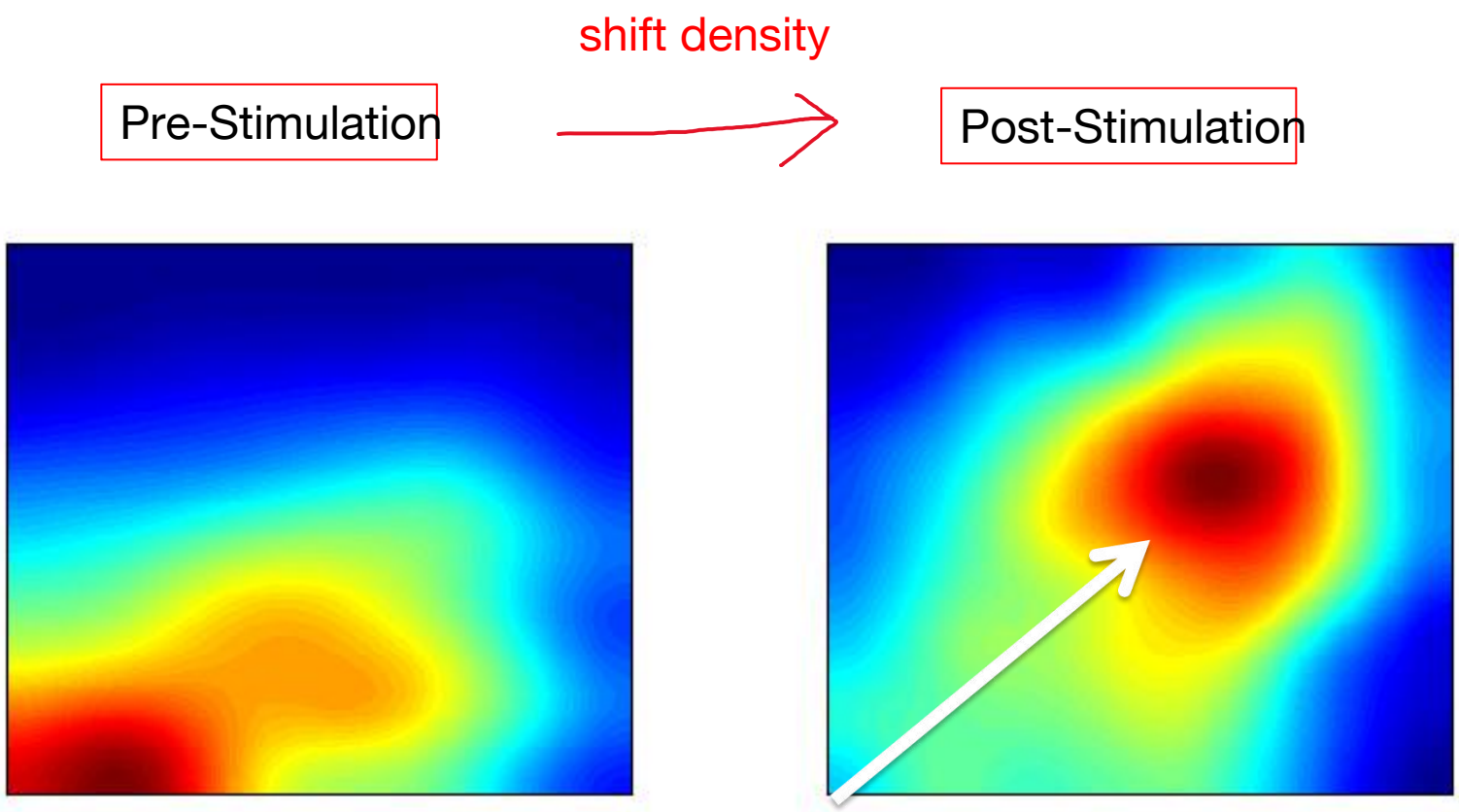
Highly varied response to stimulus

Kernel Density Estimation



KDE learns underlying probability distribution, smooths data

KDE: Reveals Activation Details



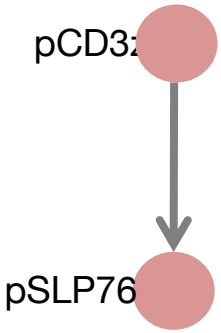
- Molecules increase together
- Shape of influence or relationship unclear

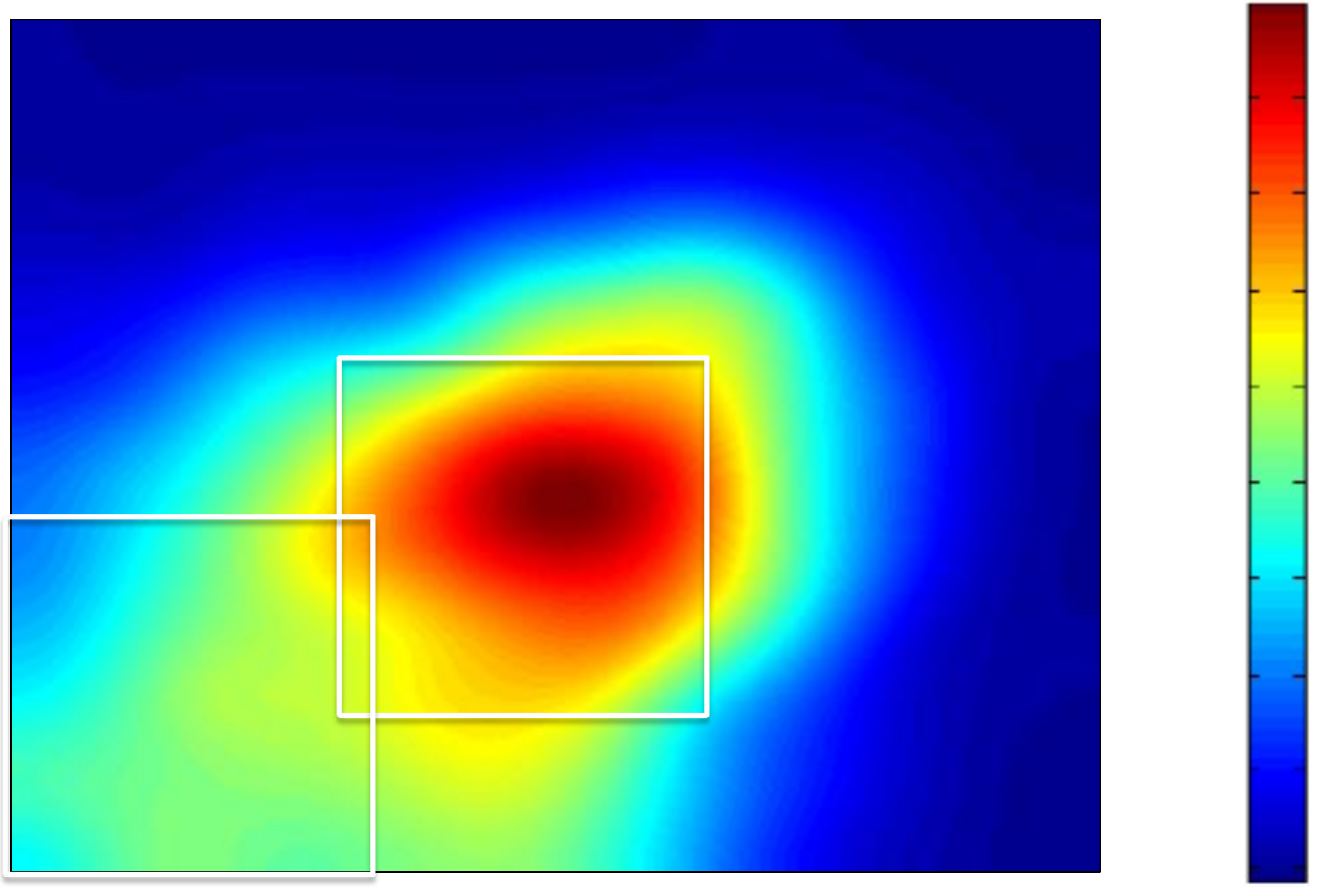
Conditional Density

$$\hat{f}_h(x \mid y) = \hat{f}_h(x, y) / \hat{f}_h(y)$$

- Main idea:
  - Obtain the 2D joint density estimate
  - Normalize the joint density by the *marginal* density

Learning in Sparser Regions

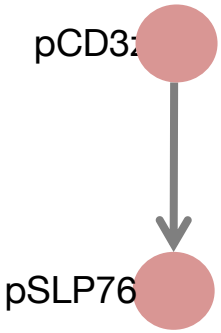


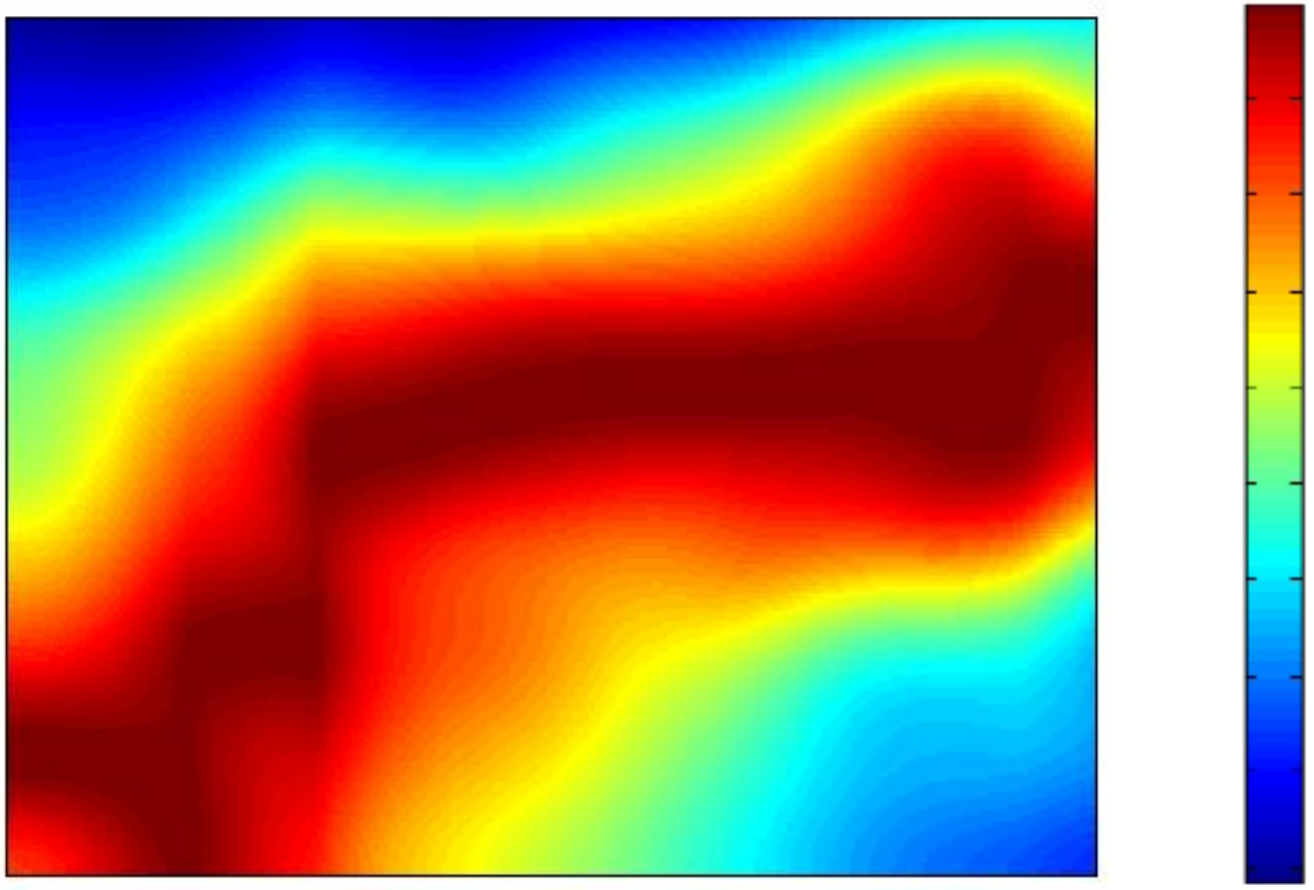


Joint distribution highlights dense areas

Account for sparse regions to learn relationship

conditional-**D**ensity **Re**scaled **VI**sual

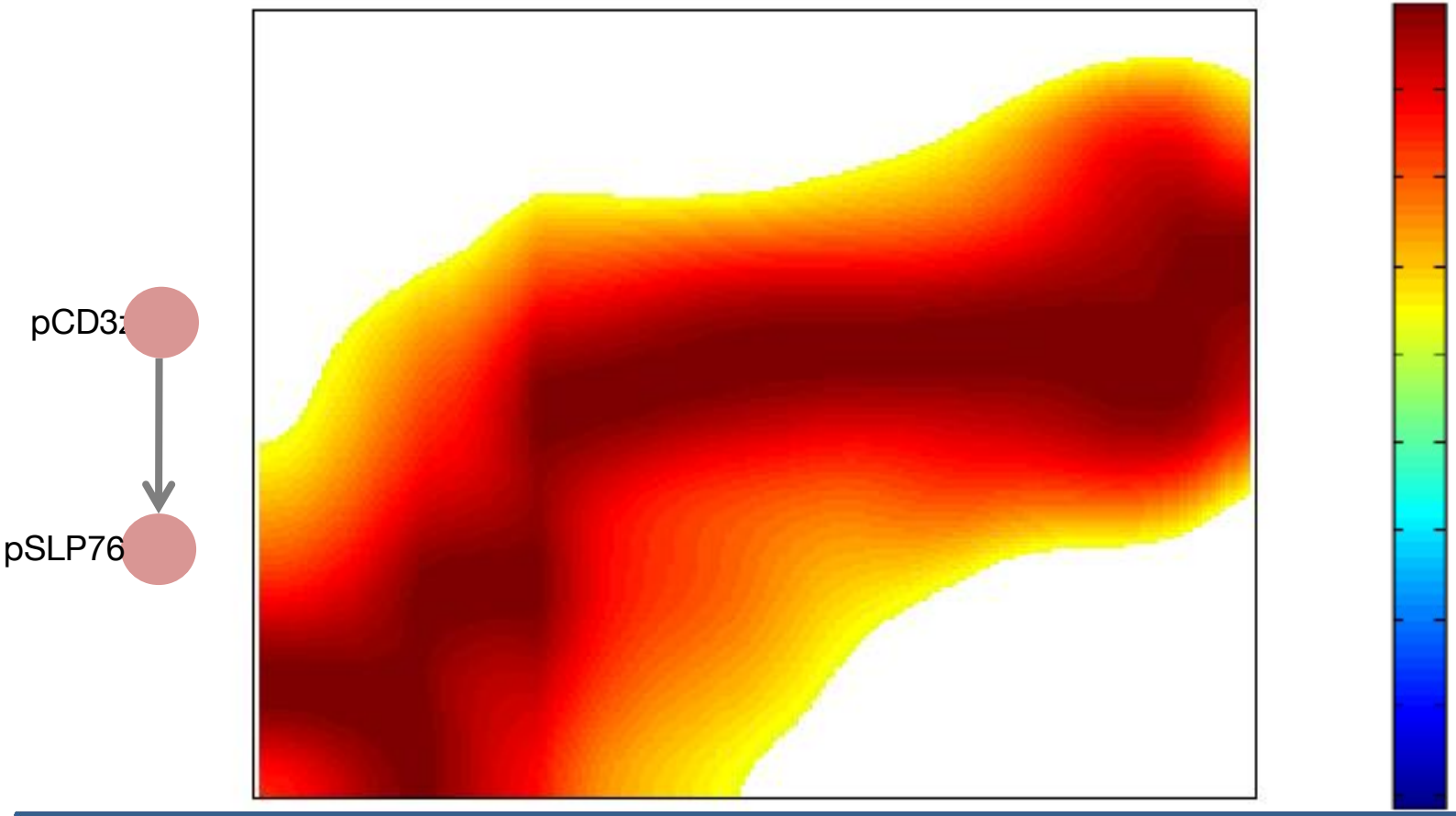




- Captures behavior across full dynamic range
- Captures behavior of small populations of responding cells

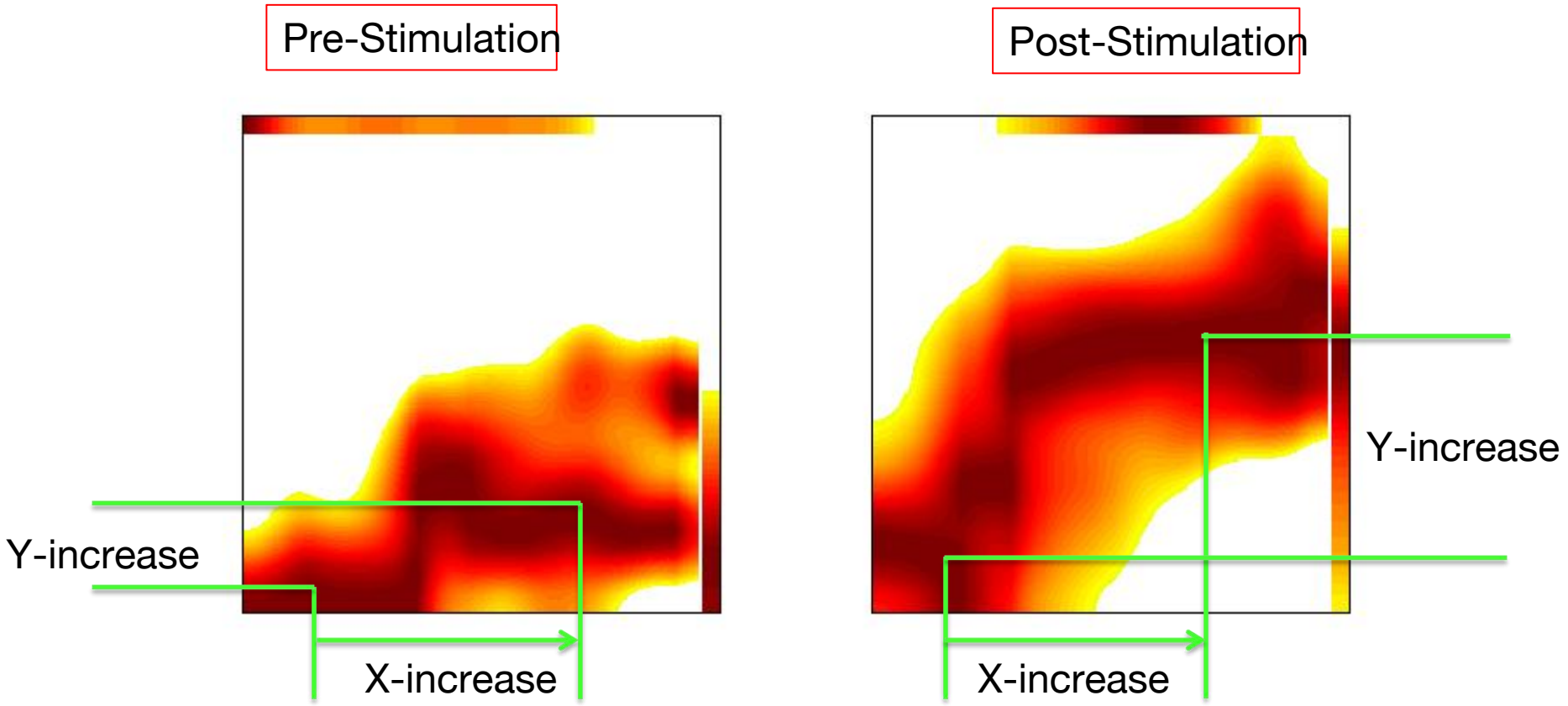
joint density focus more on dense area than sparse area,
conditional density focus evenly on areas

Adaptive Outlier-Detection



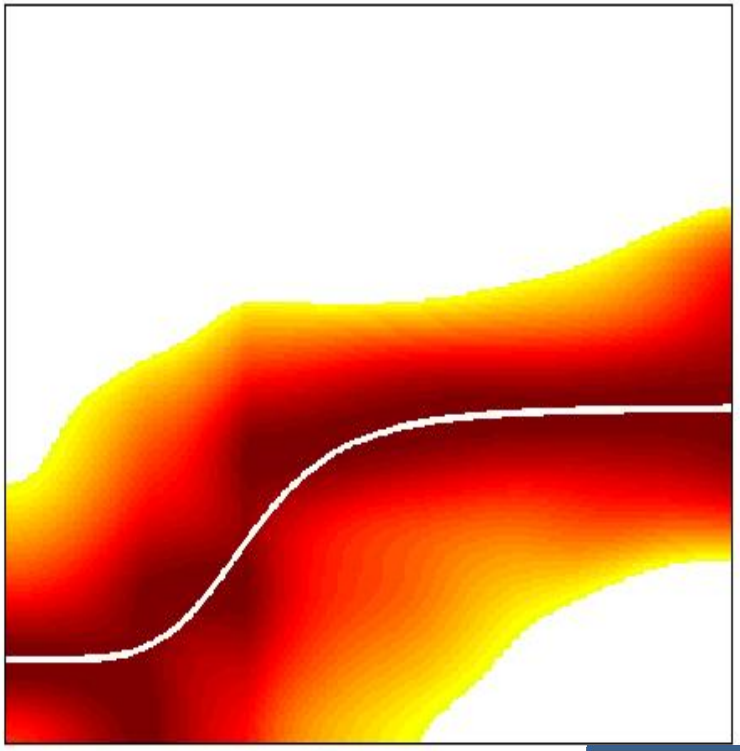
- Eliminate conditionally sparse regions to obtain sharper signal
- Filter values v< ε * peak-conditional-density

DREVI: Reveals Change in Signal Transfer Relationship



- Same increase in X leads translates to larger increase in Y
- Reconfigured relationship

Curve-Fitting Dense Region



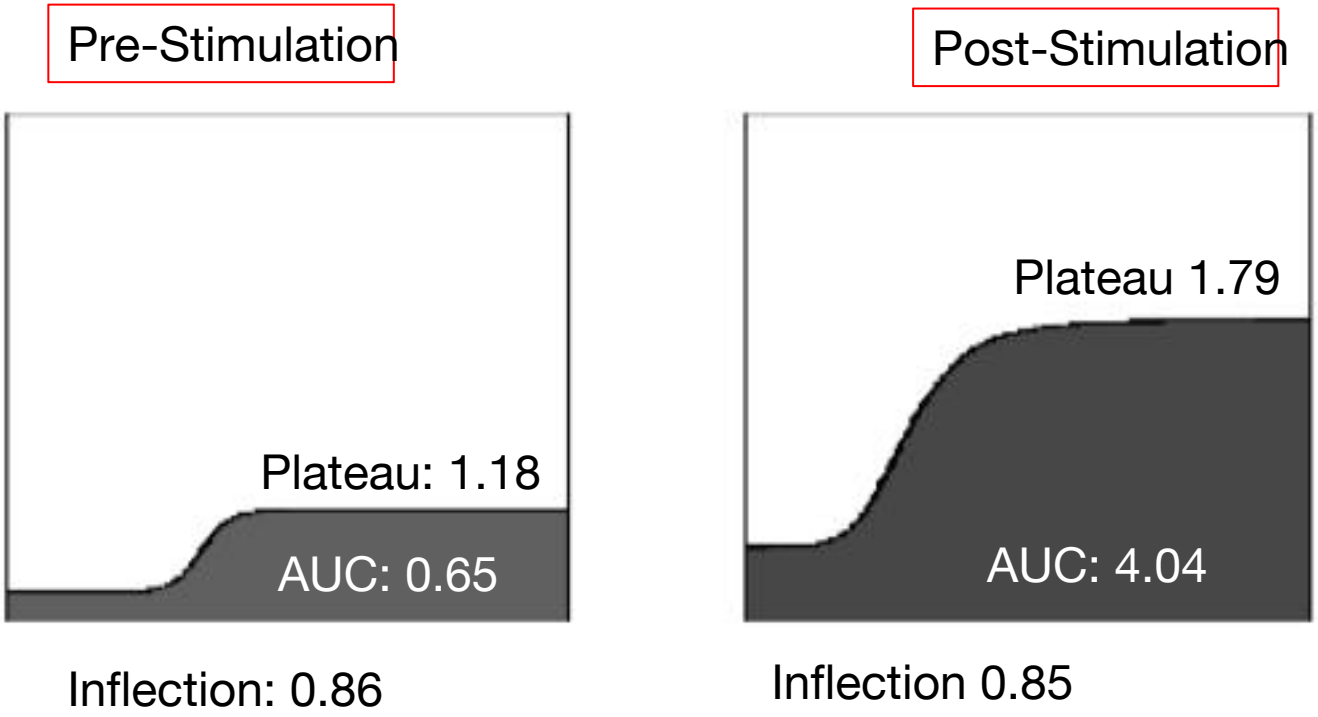
Sigmoid: $$f(x) = \frac{(A - D)}{1 + (\frac{x}{C})^{B}}$$

A= upper asymptote
B= related to slope of change
C= inflection point

Inflection (C) : 0.86
Plateau (A): 1.79
Area-Under-Curve: 4.04

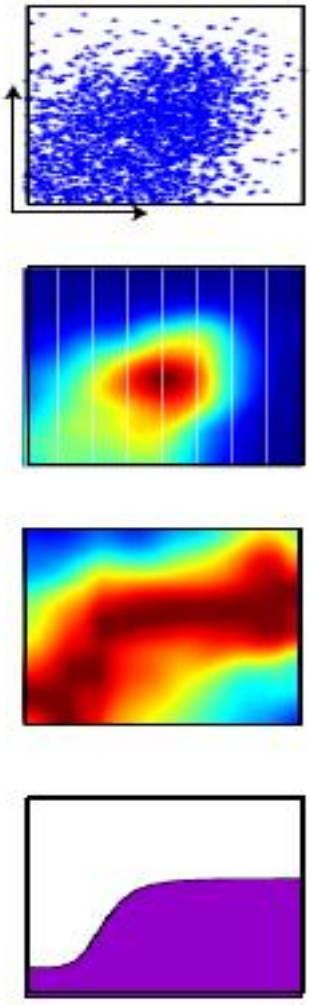
- Can potentially obtain kinetic rate parameters circuit-wide
- Linear, double-sigmoidal models also used

Regression: Parameterizes Response



Quantifies comparison

DREVI (visual)



Raw data

↓

Compute Kernel Density Estimate on a fine grid

↓

Column-normalize grid to obtain Conditional Density

↓

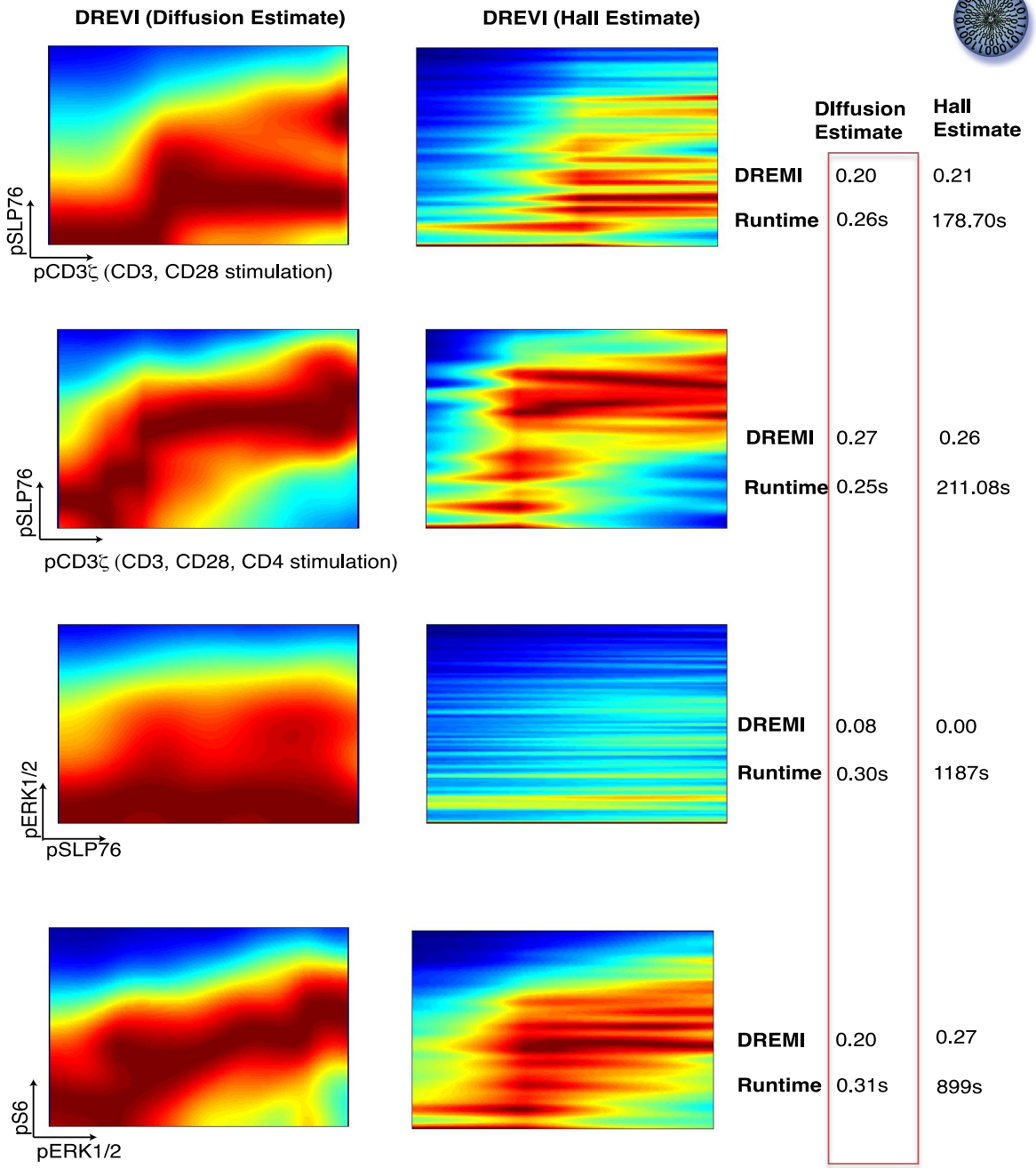
Rescale each column and render heatmap

⇣

Compute best-fit curve to high density region

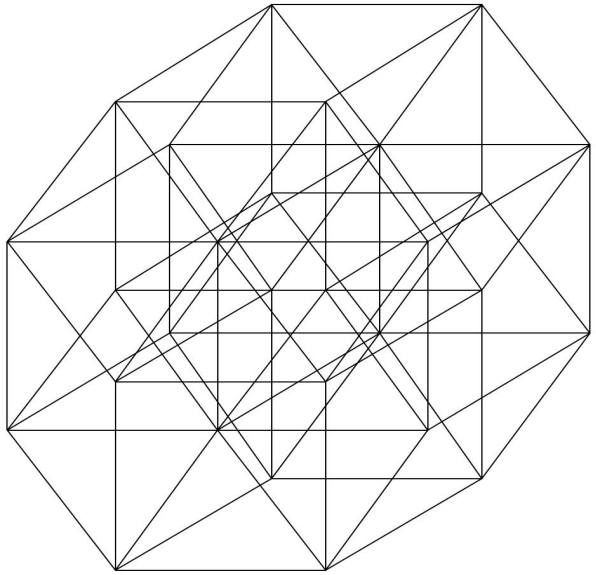
Diffusion KDE

Diffusion-based KDE estimate is faster and smoother



Need a Method of Data Exploration

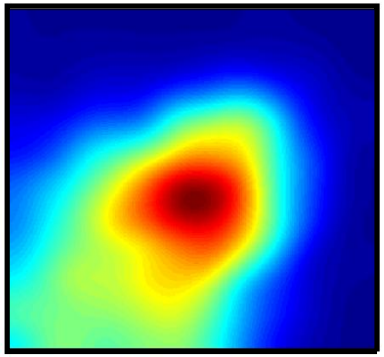
- Complex data
  - 3 Stimulation conditions
  - 2 Doses
  - 13 Time points
  - 42 proteins
  - Multiple T-cell subsets

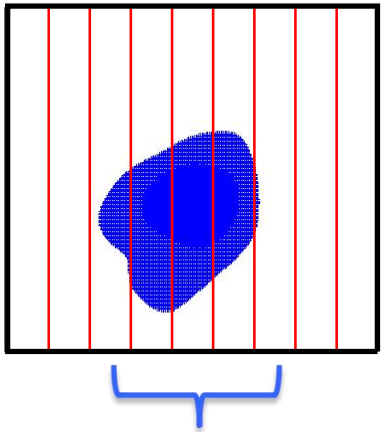


Q. Where in the data are interesting relationships?

Q. How does information flow in these dimensions?

MI is Biased by Sampling and Noise



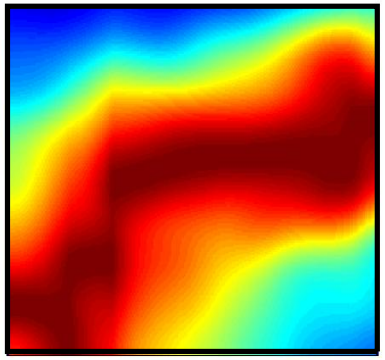


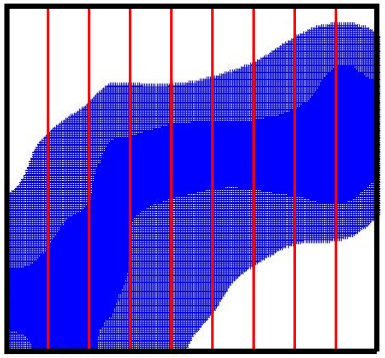
X-slice dominates MI

MI

- Majority of cells exist in a narrow band
- Score is not provided for the full relationship
- Does not account for noise explicitly

Density-Resampled Estimate of MI



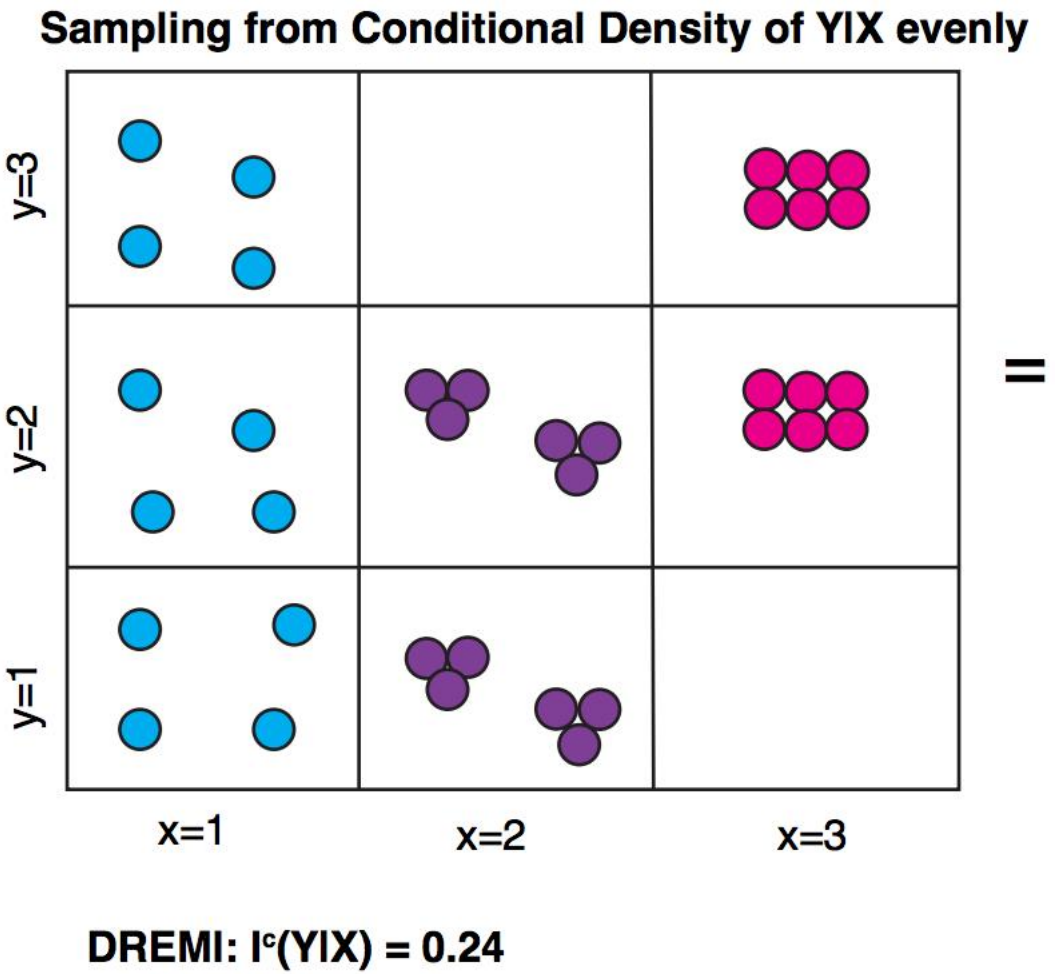


DREMI

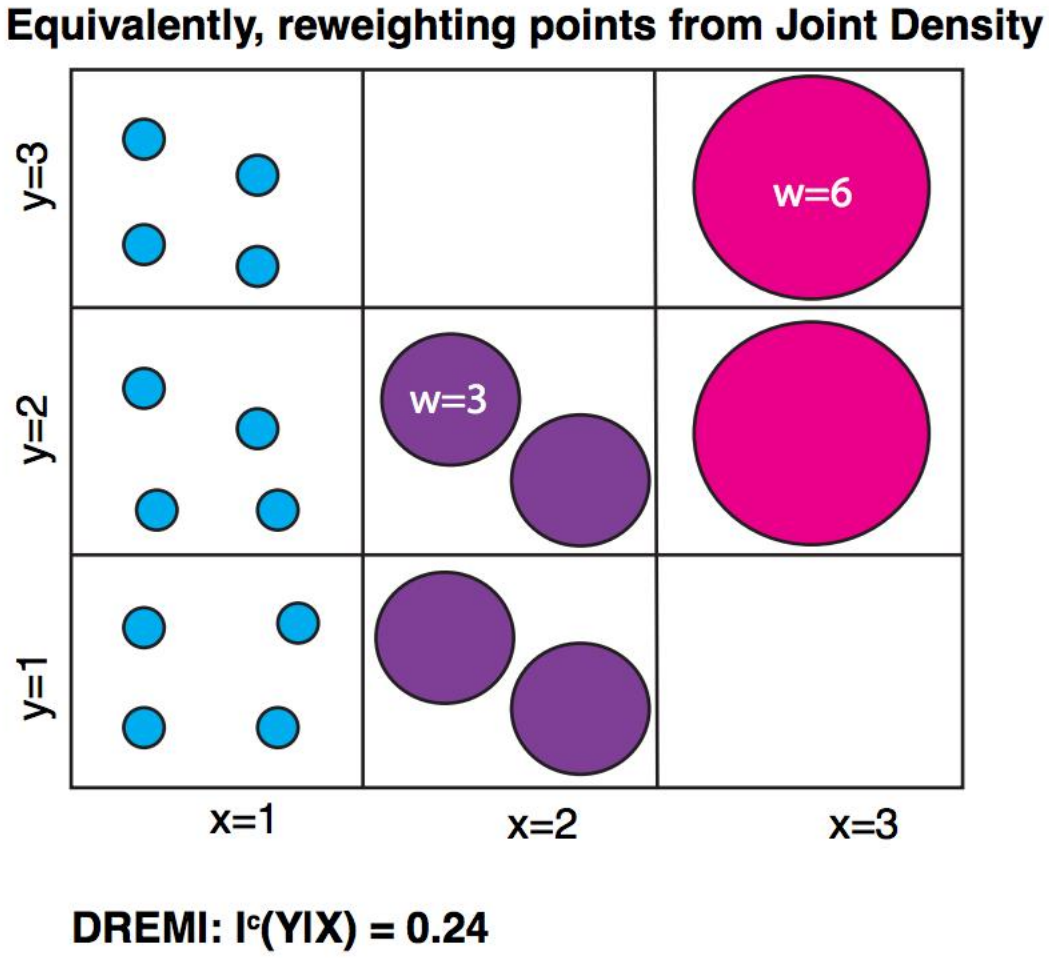
$$I^c(X,Y) = H^c(Y|X) - H^c((Y|X)|X)$$

- Computing MI on the denoised conditional density solves problems of sampling bias and noise

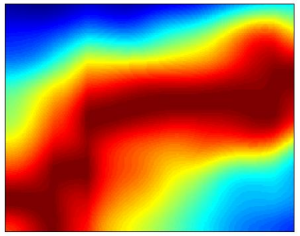
MI on Conditional Density

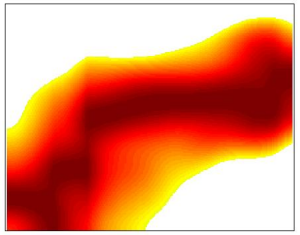


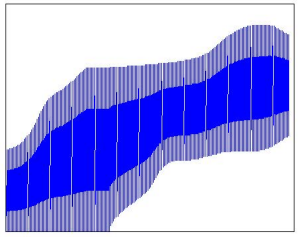
Reweighted MI

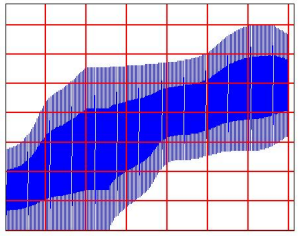


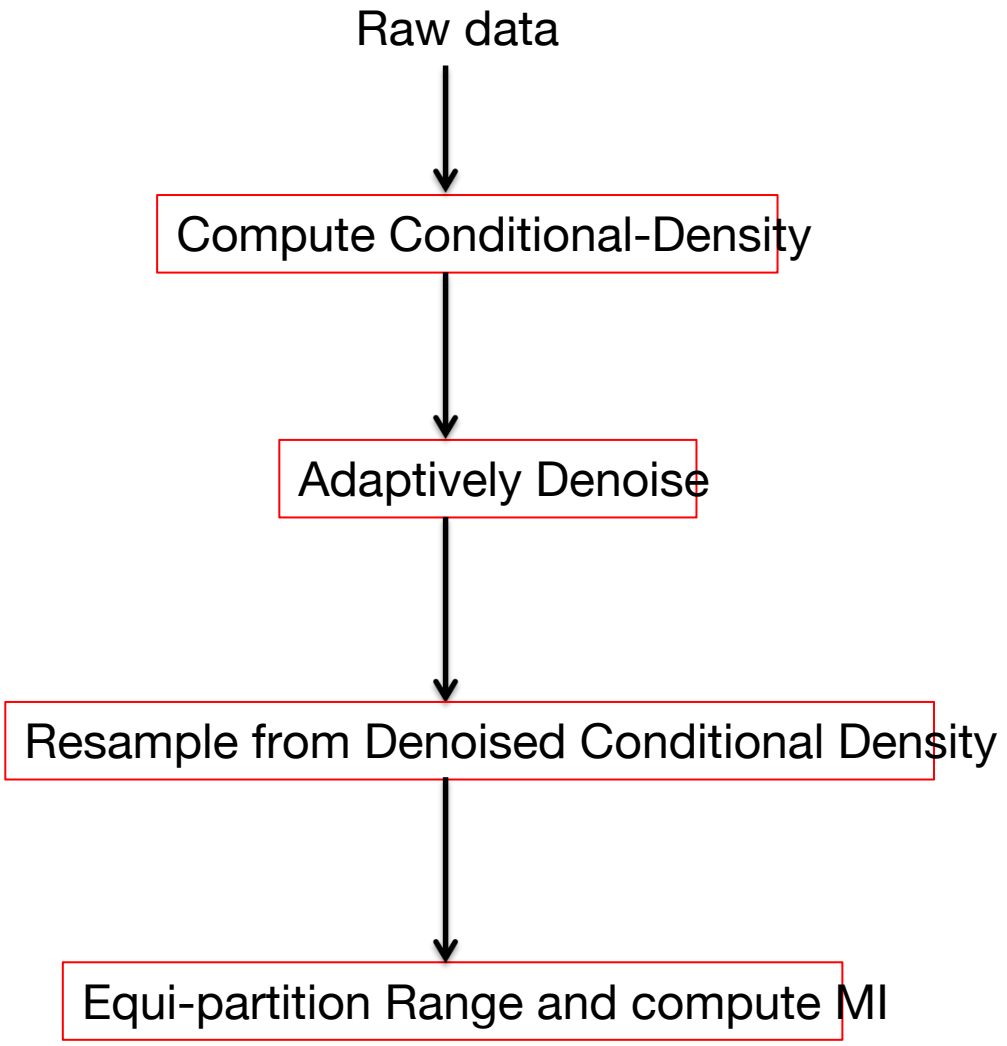
DREMI (score)



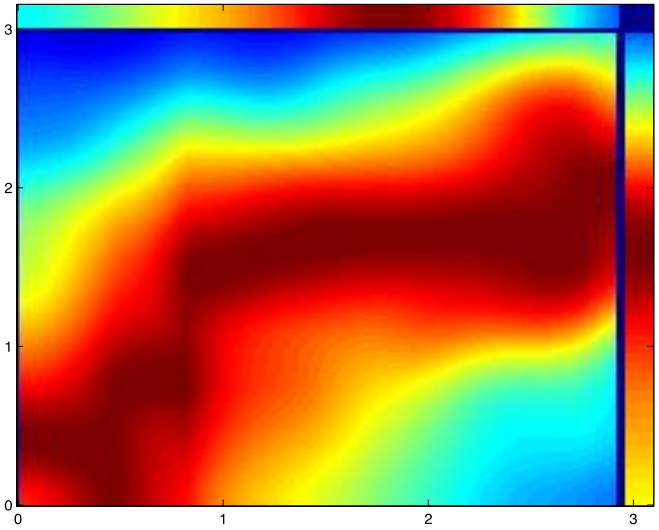




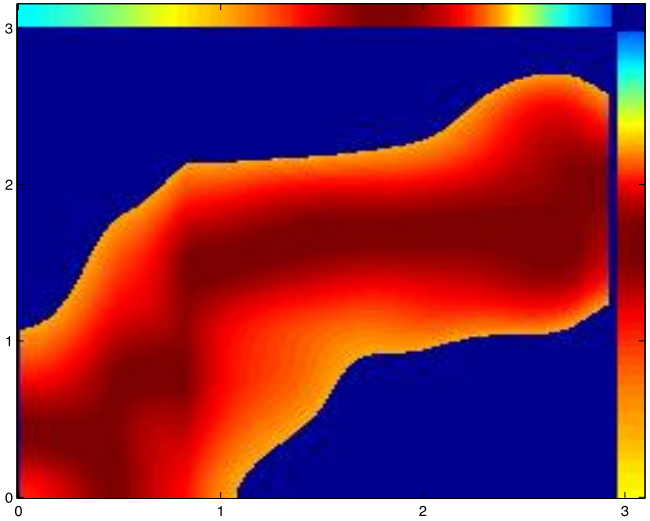



Raw data
Compute Conditional-Density
Adaptively Denoise
Resample from Denoised Conditional Density
Equi-partition Range and compute MI

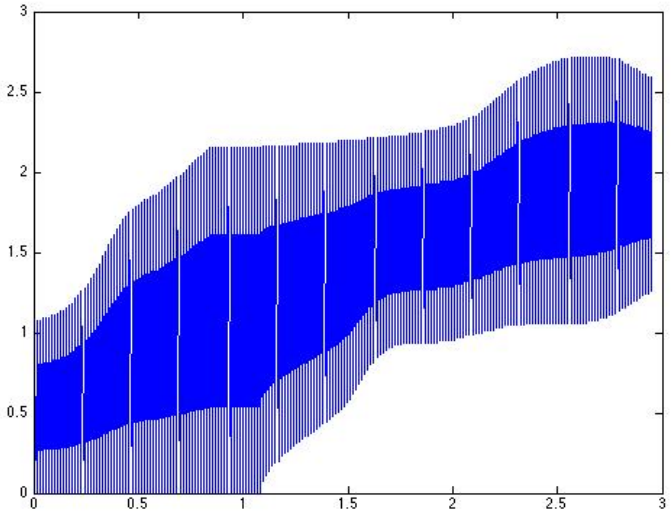
Density Reweighted estimate of Mutual Information (DReMI)



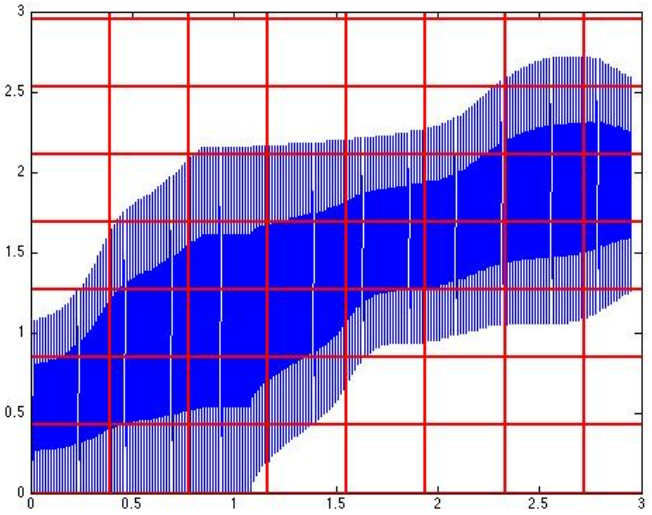
1. Start with the conditional density estimate.



2. Filter-out noise.



2. Importance sample from denoised conditional density (n samples/slice)



4. Equipartition x, y axes and Compute mutual information

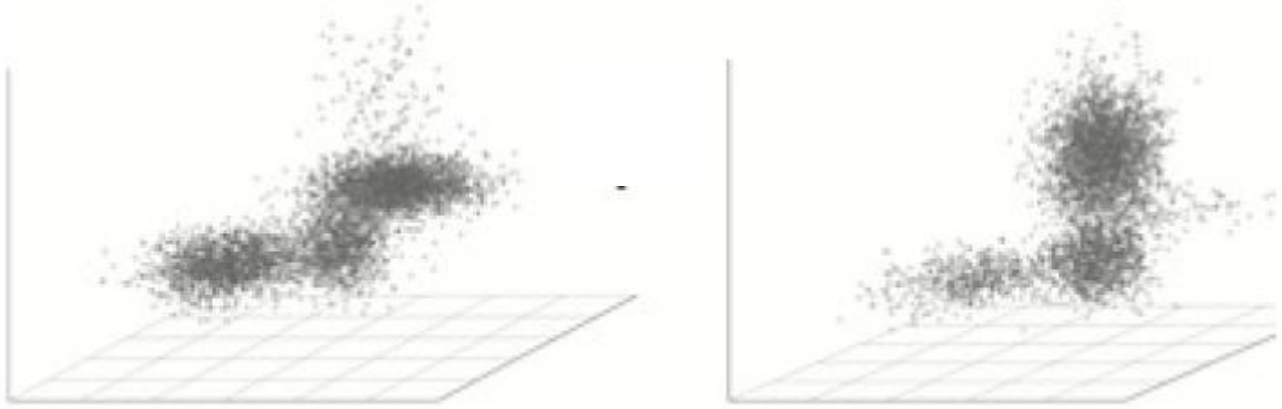
Exercise

- Explore DREMI with the given notebook

Comparing Probability Distributions: Cross and Relative Entropy

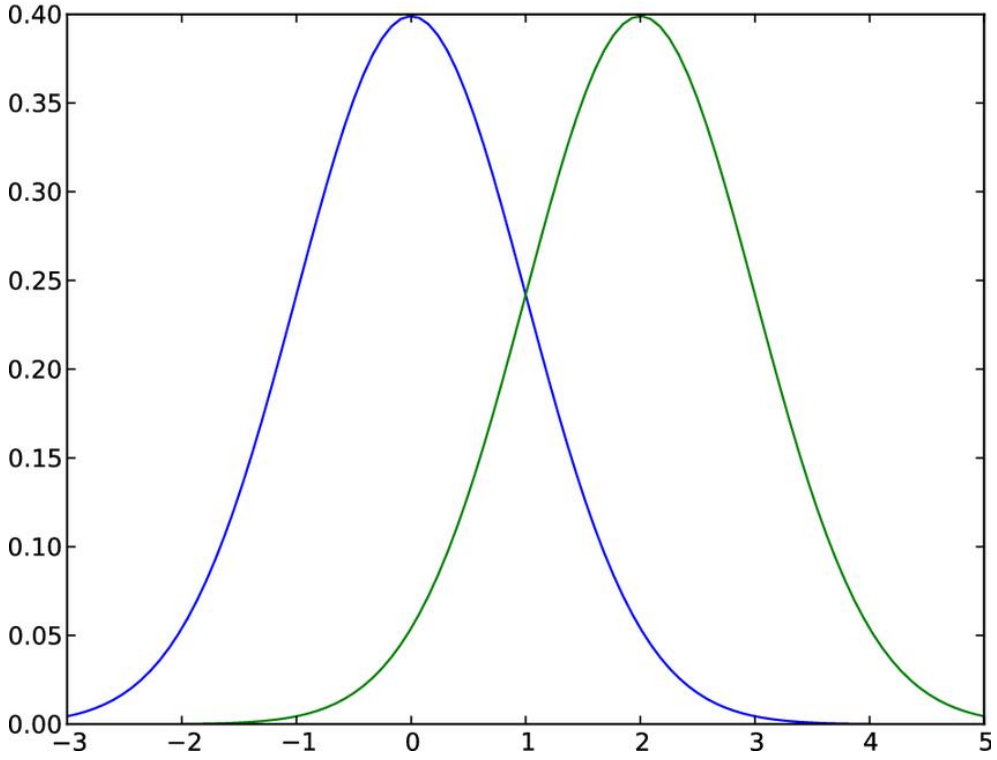
Comparing two distributions

- Point clouds, that could have amorphous shapes 无定形的
- How do you compare them?



Comparing two clouds

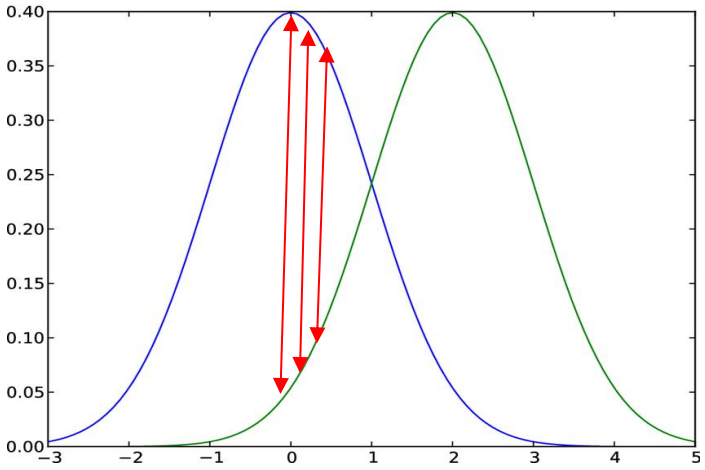
Comparing two distributions

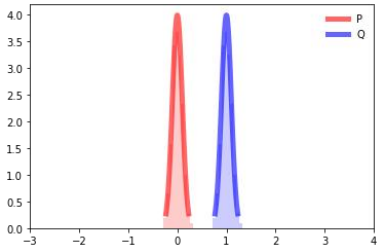


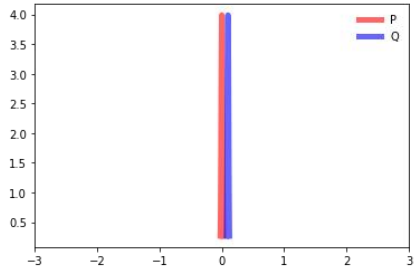
Distances

- Distances measure ways in which objects are different
- A distance metric is a real valued function d(x,y) such that
  - d(x,y)>=0 non-negative
  - d(x,x) = 0 identity
  - d(x,y)=d(y,x) symmetric
  - d(x,z)<=d(x,y)+d(y,z) triangle inequality

Total variation (pointwise) distance







Total-Variation Distance (TV)

$$\mathrm{TV}(P,Q) = \frac{1}{2}\|P - Q\|_1$$

$$\mathrm{TV}(P,Q) = 1$$

drawbacks:

- for distributions of non-overlapping, the total variation distance is always 1
- even as things get arbitrarily close

Recall: Shannon Entropy

- The expected amount of uncertainty in a distribution

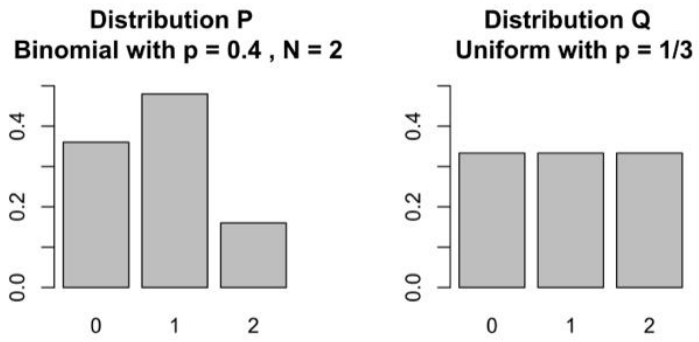
$$\mathrm{H}(X) = -\sum_{i=1}^{n} \mathrm{P}(x_i) \log_b \mathrm{P}(x_i)$$

- That is the amount of information you learn by knowing the solution
- Also, the #bits on average that it takes to transmit the solution

Cross Entropy

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

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



Encoding a distribution Q

- For instance, if you had outcomes A, B, C,D
- q(A)=1/2
- q(B)=1/4
- q(C)=1/8
- q(D)=1/8
- You could encode:
  - A as 1, B as 0
  - C as 10, D as 11
- Average number of bits needed is (3/4)*1+(1/4)*2 = 1.25

Suppose you used encoding for Q on distro P

- P is distributed differently than Q
  - p(A)=1/8
  - p(B)=1/8
  - p(C) = 3/4
  - p(D)=0
- Now what is the average number of bits needed?
- Average bits needed = (3/4)*2+(1/4)*1 = 1.75
- Using Q‘s code to transmit P takes 0.5 more bits

KL Divergence

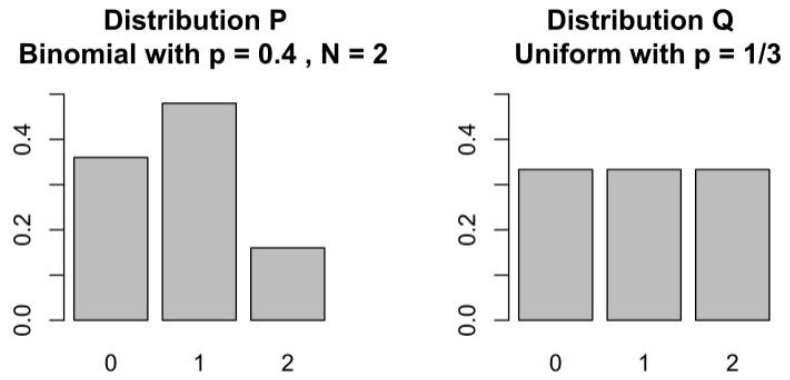
- Expected number of EXTRA bits needed if using samples from P on a code optimized for Q
- Related to cross entropy, also called *relative e*

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

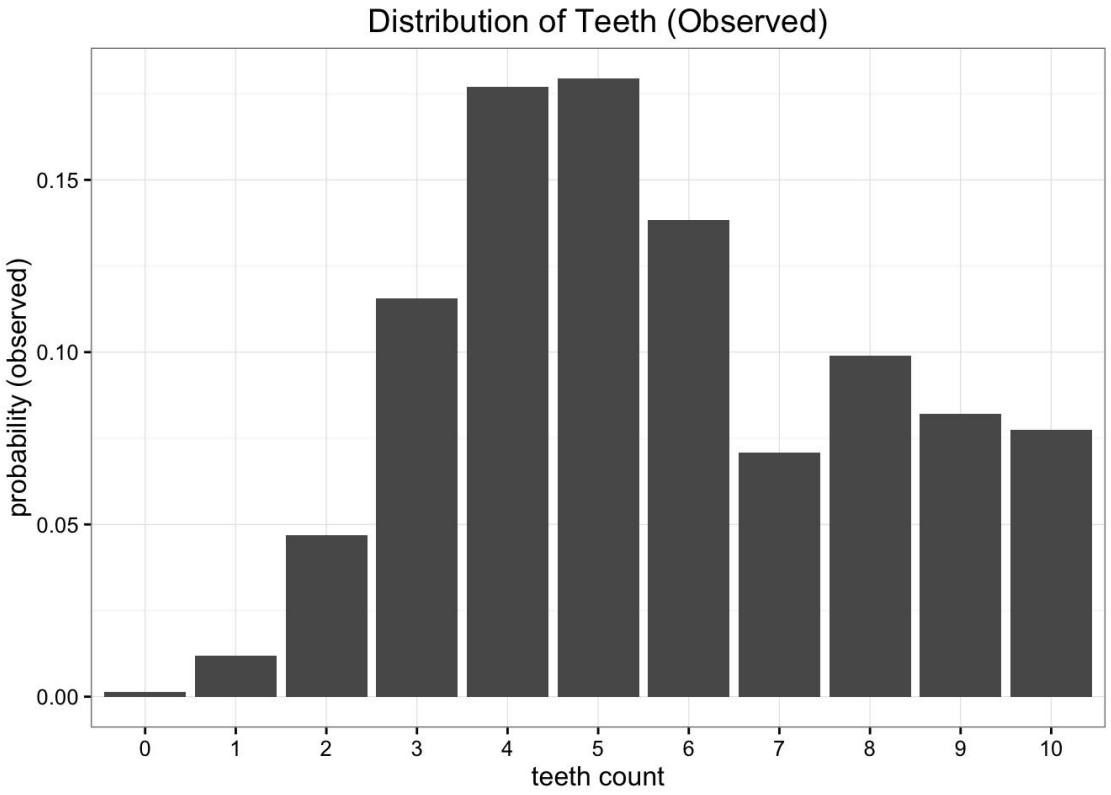
- Note that this is just H(P,Q)-H(P)

Example

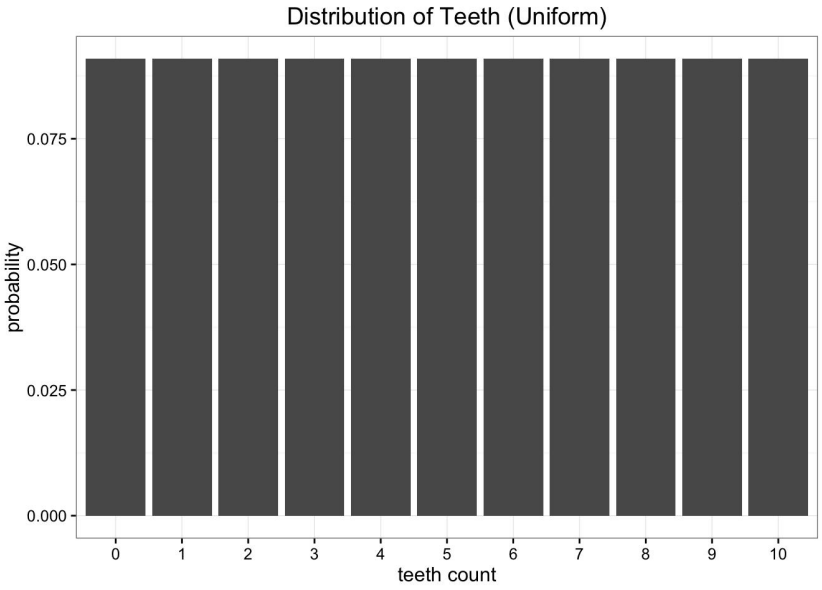
$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

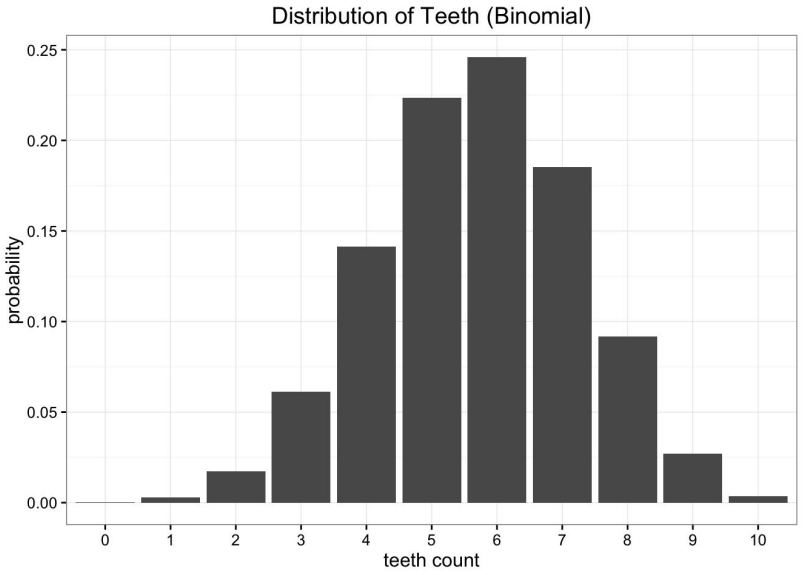


Worm teeth example



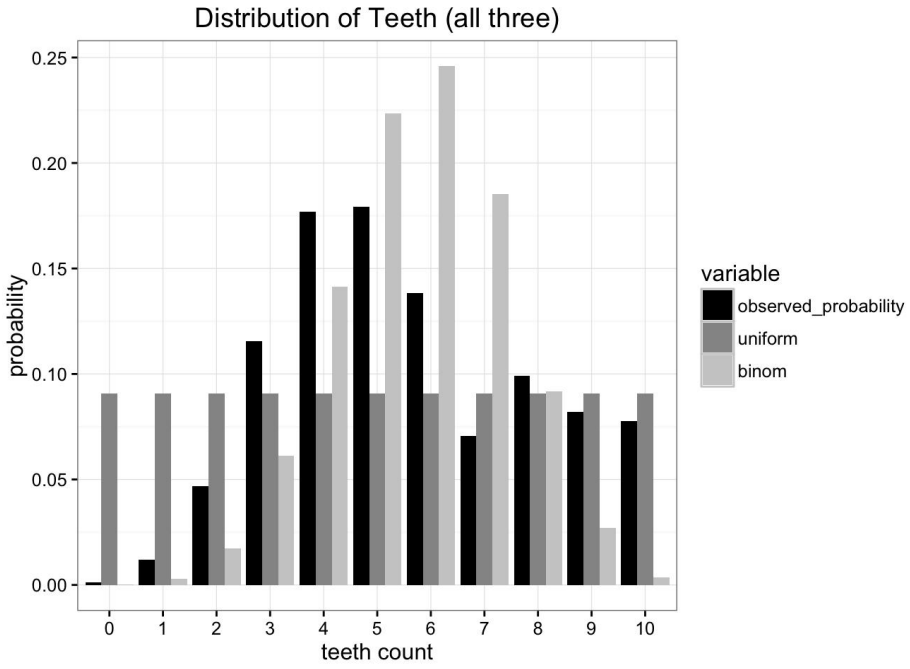
Which simple distribution better approximates this?





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

KL divergence of teeth distribution



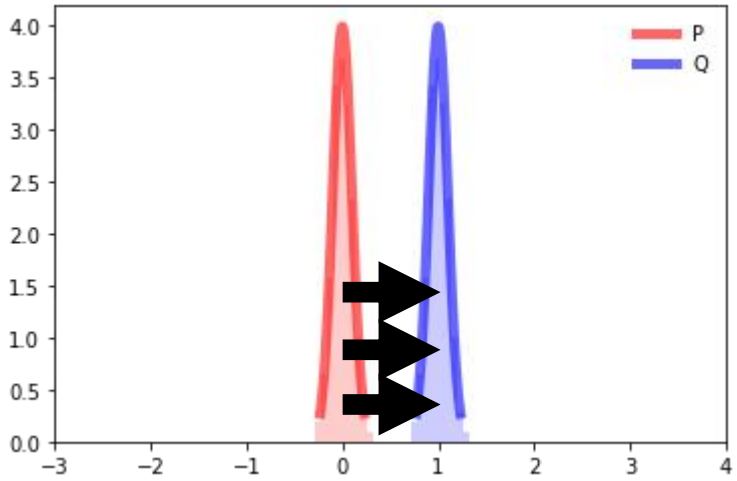
$$D_{kl}(\text{Observed} \,||\, \text{Uniform}) = 0.338$$

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

KL Divergence is not a distance

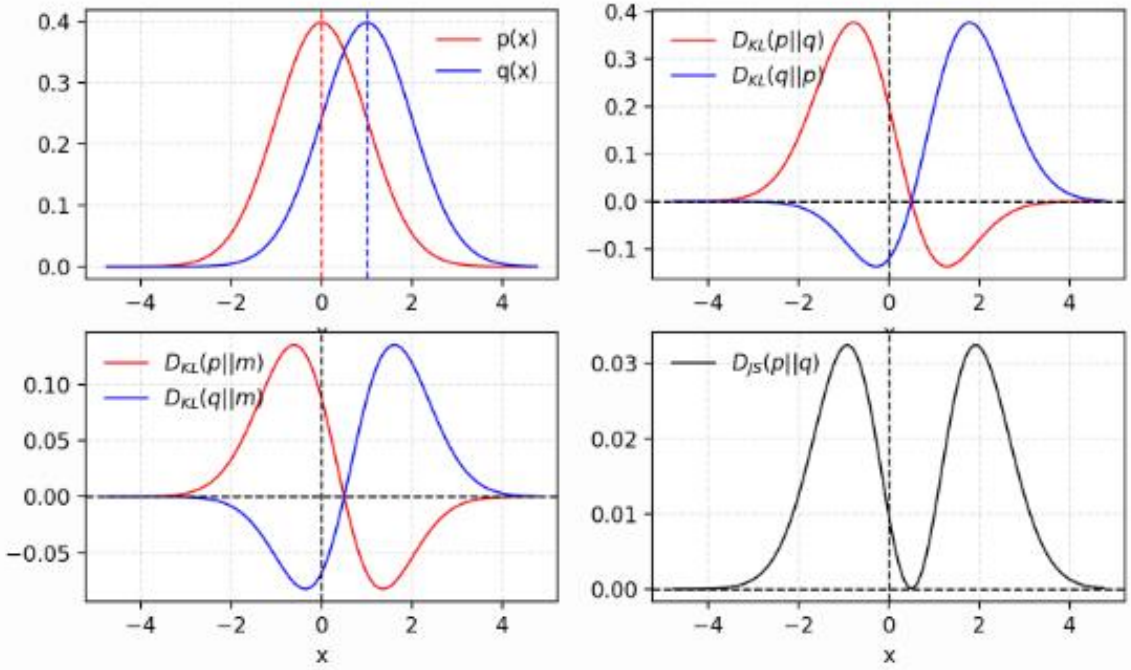
$$D_{\mathrm{KL}}(P \parallel Q) = -\sum_{x \in \mathcal{X}} P(x) \log\left(\frac{Q(x)}{P(x)}\right)$$

- Asymmetric: D(P||Q) is not the same as D(Q||P)
- It also does not follow the triangle inequality!
- problem: KL(P || Q) of non-overlap distributions = infinity

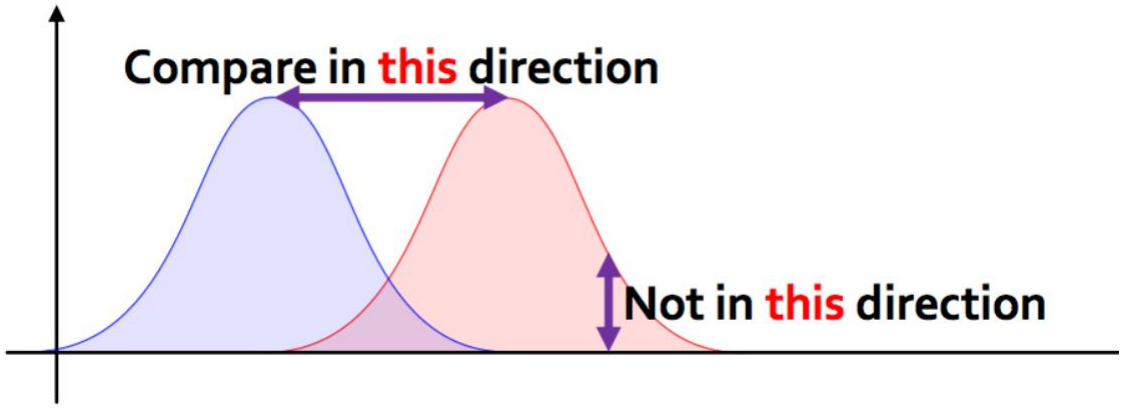


Jensen-Shannon Divergence

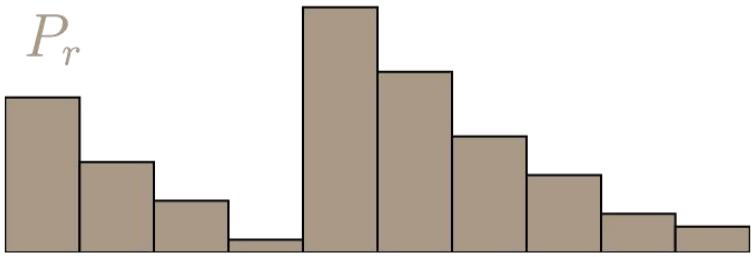
$$D_{JS}(p\|q) = \frac{1}{2}D_{KL}(p\|\frac{p+q}{2}) + \frac{1}{2}D_{KL}(q\|\frac{p+q}{2})$$

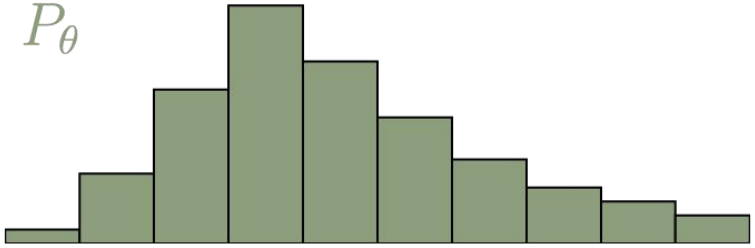


Key insight: Ground distance



Transportation Cost





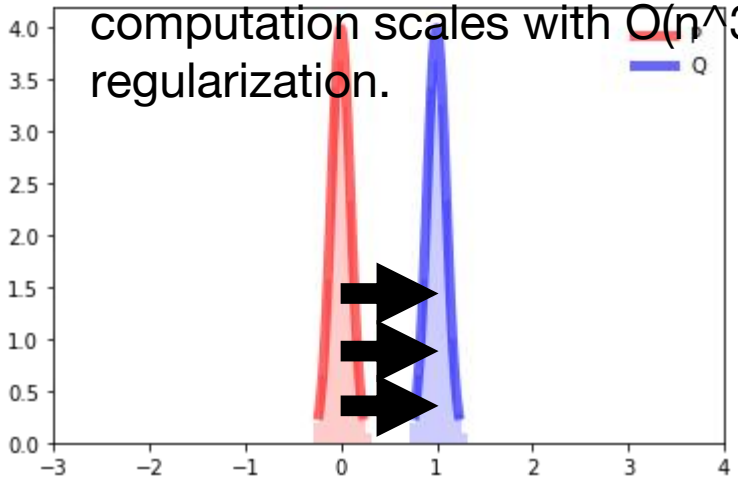
Optimal Transport – The Earth Mover's Distance

Another distance between probability distributions is the Earth Mover's Distance, also known as the Wasserstein distance.

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

Intuitively, this distance measures the amount of work it would take to move earth piled at one distribution to another to piles defined by another distribution.

This earth mover's distance does not suffer from the problem described previously, but it suffers from computational challenges. In the primal form this computation scales with O(n^3) for exact solutions and O(n^2) using entropic regularization.



Wasserstein Distance:

$$W_d(P,Q) = \inf_{\pi \in \prod(P,Q)} \int d(x,y)\pi(dx,dy)$$

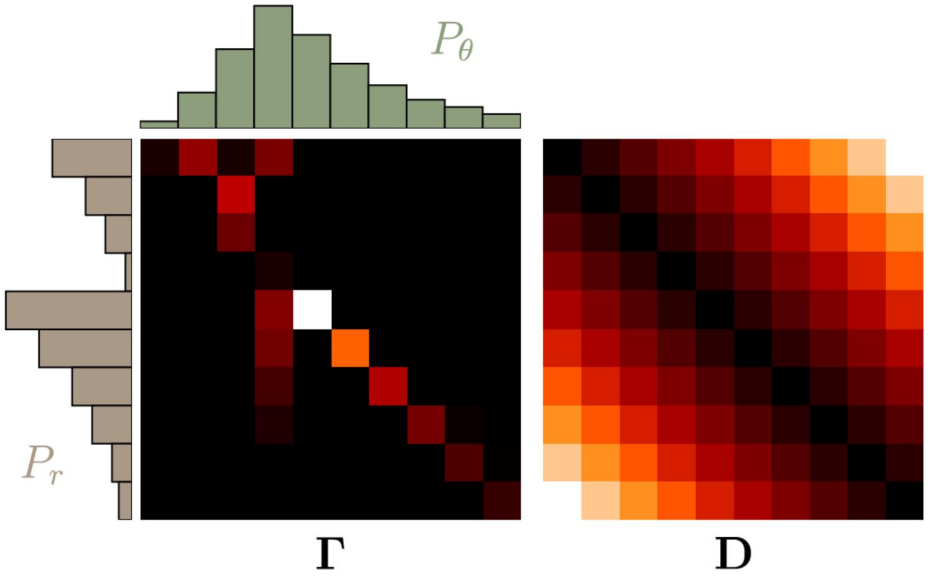
Ground "cost" or distance:

$$d(x,y) = \|x - y\|_2$$

“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

Optimal Transport Plan



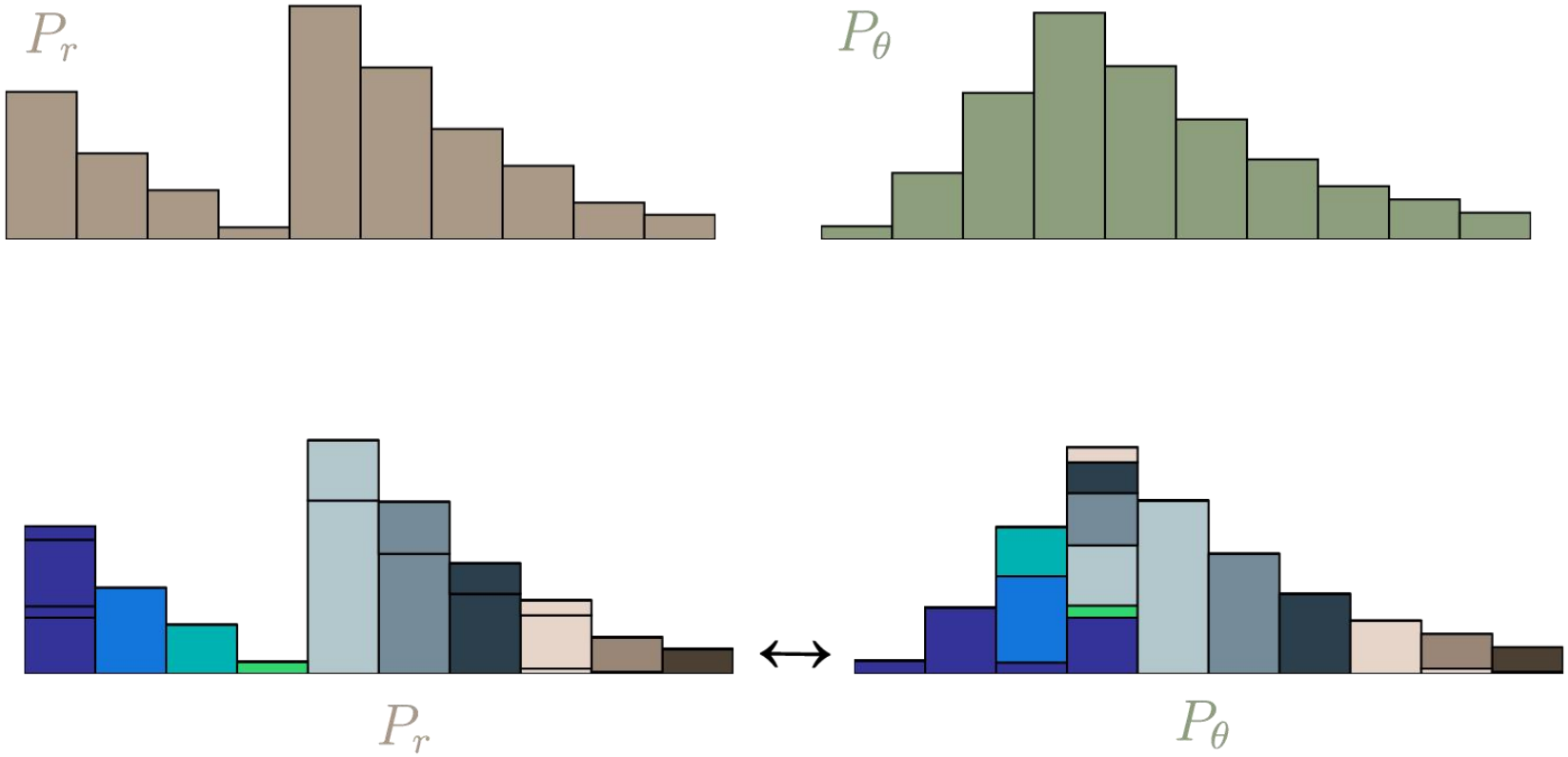
Minimize

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

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

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

Optimal Transport Plan



Exercise

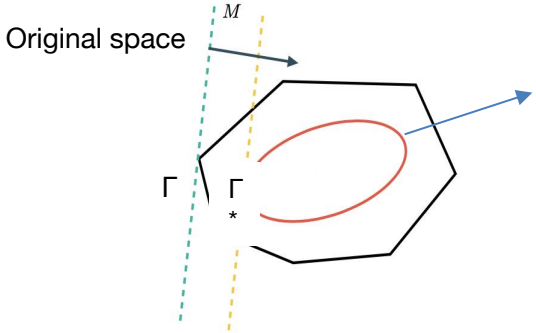
- Why is EMD a distance?

Entropic regularization

- Add an entropic regularization term, if there was no cost, we'd spread mass evenly

$$c = \sum_{(x,y)} \Gamma_{x,y} D_{x,y} - \frac{1}{\lambda} H(\Gamma) \qquad H(\Gamma) = -\sum_{(x,y)} \Gamma_{x,y} \log(\Gamma_{x,y})$$

- Makes a more stochastic transport, may be useful in some circumstances
- Creates a much faster easier optimization!



Level surfaces of entropy are smooth in probability Distribution space

Solution has to have the form: $\Gamma_{(x,y)} = \alpha_x \beta_y e^{-\lambda D_{(x,y)}}$

Relation to Boltzman (Gibbs) Distribution

- The Boltzman distribution is a distribution from the field of statistical mechanics where the probability that the system in a certain state is a function of that state's energy and temperature of the system

$$p_i \propto e^{-\frac{\varepsilon_i}{kT}}$$

- p_i is the probability of the system being in state i
  - negative exponential dependence on the energy of that state (states with higher energy have lower probability)
  - mitigated by the overall temperature of the system

$$\Gamma_{(x,y)} = e^{-\lambda D_{x,y}}$$

- Here λ is the inverse of the temperature, and D is the "energy"

Sinkhorn iteration algorithm

- Given D P_r P_θ λ
- Initialize $\Gamma_{(x,y)} = e^{-\lambda D_{x,y}}$
- Repeat
  - Rescale rows of Γ so that

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

  - Rescale cols of Γ so that

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

EMD optimization

Minimize $\sum_{x,y} \|x - y\| \gamma(x, y)$

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

Dual of a linear program

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 $z = \tilde{z}$

Linear Program Formulation

c is a vectorized form of D:

$$\left[\begin{array}{ccc|ccc|c|ccc} \mathbf{D}_{1,1} & \mathbf{D}_{1,2} & \dots & \mathbf{D}_{2,1} & \mathbf{D}_{2,2} & \dots & \dots & \mathbf{D}_{n,1} & \mathbf{D}_{n,2} & \dots \end{array}\right] \Big\} \mathbf{c}^T$$

A sums up the correct entries of x

$$\left[\begin{array}{ccc|ccc|c|ccc} 1 & 1 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots \\ 0 & 0 & \dots & 1 & 1 & \dots & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & \dots & 1 & 1 & \dots \\ \hline 1 & 0 & \dots & 1 & 0 & \dots & \dots & 1 & 0 & \dots \\ 0 & 1 & \dots & 0 & 1 & \dots & \dots & 0 & 1 & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \dots & \vdots & \vdots & \ddots \\ 0 & 0 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots \end{array}\right] \Big\} \mathbf{A}$$

Strong Duality.

- Under the strong duality theorem, the primal and dual have the same solution
- Therefore $\tilde{z}^* = \mathbf{b}^T \mathbf{y}^*$ is the EMD
- Let $\mathbf{y}^* = \begin{bmatrix} \mathbf{f} \\ \mathbf{g} \end{bmatrix}$
- Thus $$\mathrm{EMD}(P_r, P_\theta) = \mathbf{f}^T P_r + \mathbf{g}^T P_\theta$$
-
- Also since P_r and P_θ are positive

$$\sum_i \mathbf{f}_i + \mathbf{g}_i$$

Dual form

$$\mathbf{A}^T\mathbf{y} \leq \mathbf{c} \qquad f(x_i) + g(x_j) \leq \mathbf{D}_{i,j}$$

$$\mathbf{y}\left\{\begin{bmatrix} f(x_1) \\ f(x_2) \\ \vdots \\ f(x_n) \\ \hline g(x_1) \\ g(x_2) \\ \vdots \\ g(x_n) \end{bmatrix}\right. \quad \left.\begin{bmatrix} 1 & 1 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots \\ 0 & 0 & \dots & 1 & 1 & \dots & \dots & 0 & 0 & \dots \\ \vdots & \vdots & \vdots & \vdots & \vdots & \vdots & \dots & \vdots & \vdots & \vdots \\ 0 & 0 & \dots & 0 & 0 & \dots & \dots & 1 & 1 & \dots \\ \hline 1 & 0 & \dots & 1 & 0 & \dots & \dots & 1 & 0 & \dots \\ 0 & 1 & \dots & 0 & 1 & \dots & \dots & 0 & 1 & \dots \\ \vdots & \vdots & \ddots & \vdots & \vdots & \ddots & \dots & \vdots & \vdots & \ddots \\ 0 & 0 & \dots & 0 & 0 & \dots & \dots & 0 & 0 & \dots \end{bmatrix}\right\} \mathbf{A}$$

The case $i = j$ yields $g(x_i) \leq -f(x_i)$ for all i, because $\mathbf{D}_{i,i} = 0$

Dual Form of EMD

$\sum_i \mathbf{f}_i + \mathbf{g}_i$ Has to achieve maximum value, but since its sum is 0 at maximum

Therefore the solution has to have the form:

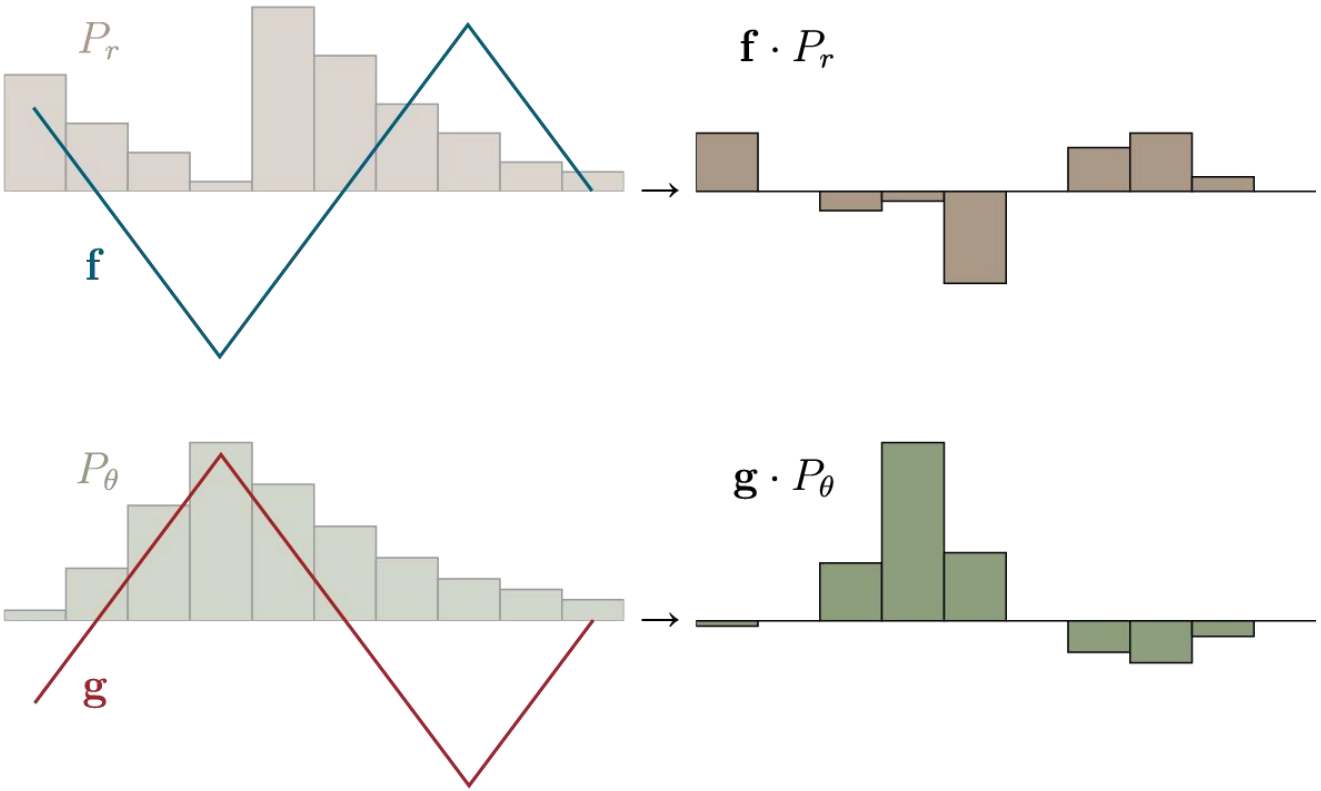
$$g = -f.$$

For $g = -f$ the constraints become $f(x_i) - f(x_j) \leq \mathbf{D}_{i,j}$ and $f(x_i) - f(x_j) \geq -\mathbf{D}_{i,j}$

This is called a lipschitz continuity constraint (the function f can't change too fast).

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

F is also called a witness function



Kantorovich Rubenstein Duality

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

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

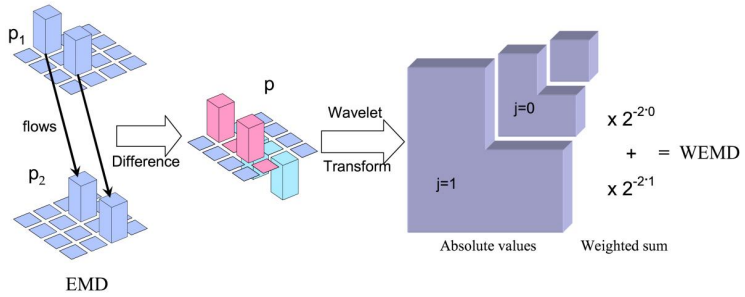
$$= \sup_{\|f\|_{L \leq 1}} \mathbb{E}_{s \sim p_r} [f(s)] - \mathbb{E}_{t \sim p_\theta} [f(t)]$$

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

Does not give a transport plan!

Multiscale histograms/Density estimates
Computing EMD with the Dual Form

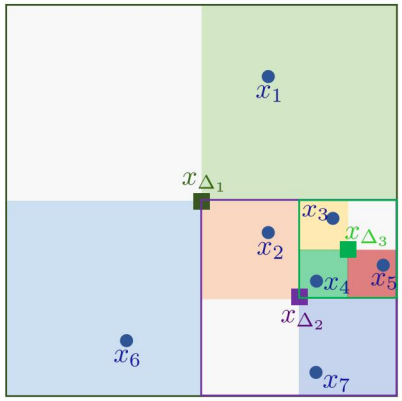
- Take the difference of two histograms at different scales
- WEMD uses a wavelet basis to represent the difference histogram
- Since wavelets are a rich basis the wavelet transform is an approximation of f
- We recently had a paper on Diffusion EMD that does this on a graph [Tong et al. ICML 2021]

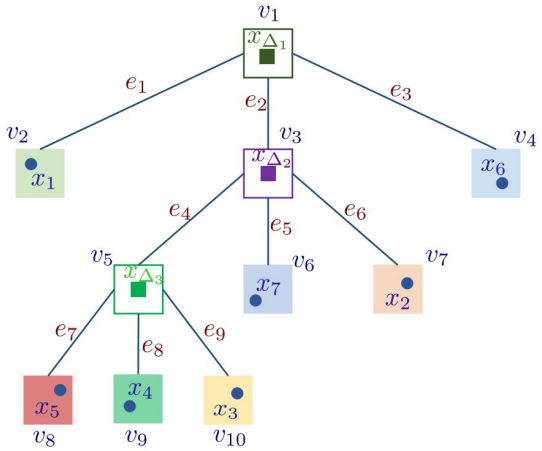


$$d(p)_{wemd} = \sum_{\lambda} 2^{-j(1+n/2)} |p_\lambda|$$

Tree-sliced Wasserstein

- Suppose you have data from 2 distributions P, Q
- Combine the data and approximate the data with a tree
  - Recursive bisection partition (for instance)





L1 difference between nodes

Proposition 1. *Given two measures μ, ν supported on $\mathcal{T}$, and setting the ground metric to be $d_{\mathcal{T}}$, then*

$$W_{d_{\mathcal{T}}}(\mu,\nu) = \sum_{e\in\mathcal{T}} w_e \left|\mu(\Gamma(v_e)) - \nu(\Gamma(v_e))\right|. \tag{3}$$

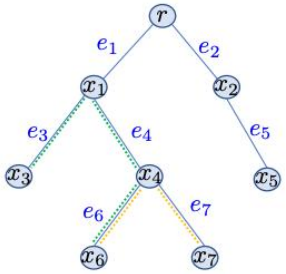
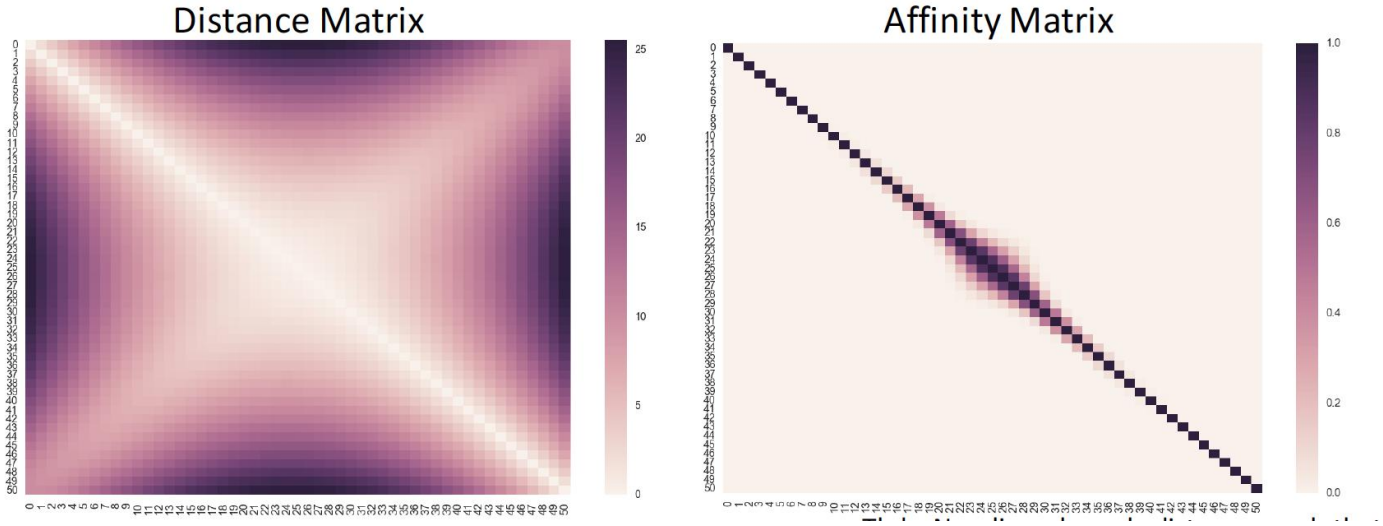


Figure 1: An illustration for a tree with root r where x_1, x_2 are at depth level 1, and x_6, x_7 are at depth level 3. Path $\mathcal{P}(x_3, x_6)$ contains e_3, e_4, e_6 (the green-dot path), $\Gamma(x_4) = \{x_4, x_6, x_7\}$ (the yellow-dot subtree), $v_{e_4} = x_4$, and $u_{e_4} = x_1$.

Density estimate necessary?

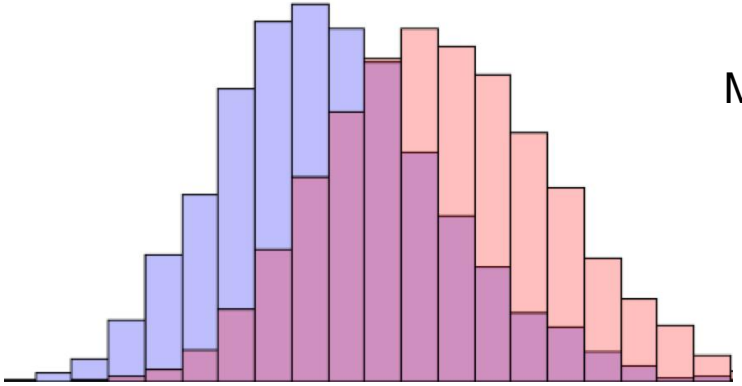
- Do we need the density estimate to compute a distance between distributions?
- It turns out that we don’t need an explicit density estimate
- We can actually use an affinity kernel defined on data to compute a distribution distance

Distance->Affinity



$$sij := \exp\left(-\frac{d(x_i, x_j)^2}{2\sigma^2}\right)$$

Maximum Mean Discrepancy (MMD)



MMD quantifies, how similar are two sets of samples by

- Picking a pair X, X' from distribution 1
- Picking a pair Y, Y' from distribution 2
- compare within-sample pairs X, X' and Y, Y to across-sample pairs X,Y and X',Y'
- calculate average distance between the within sample similarities and across samples similarities

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

[Gretton 2012]

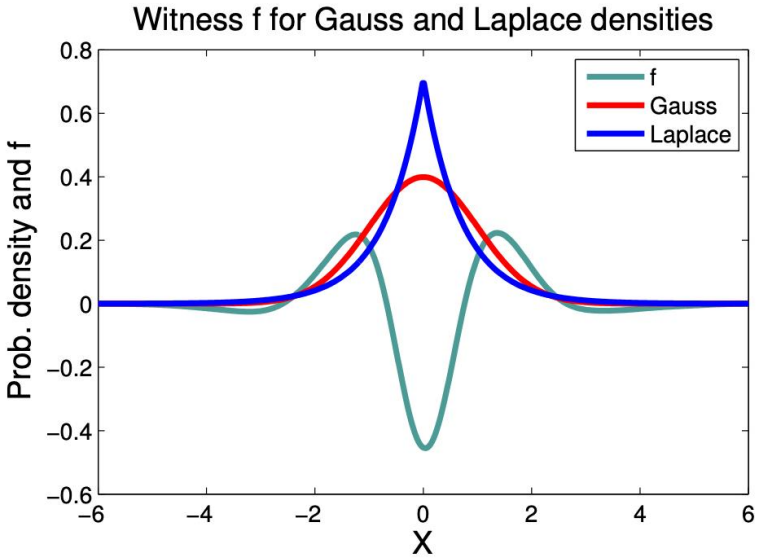
Maximal mean discrepancy

MMD is a special implementation of the dual that uses the Kernel Trick

$$\mathrm{MMD}(\mathbf{P}, \mathbf{Q}; F)$$

$$= \sup_{f \in F} \left[\mathbf{E}_{\mathbf{P}} f(\mathrm{x}) - \mathbf{E}_{\mathbf{Q}} f(\mathrm{y}) \right]$$

supremum of a set is its least upper bound



The mean trick (courtesy Arthur Gretton)

The reproducing property (kernel trick)

- Given $x \in \mathcal{X}$ for some set $\mathcal{X}$, define feature map $\varphi(x) \in \mathcal{F}$,

$$\varphi(x) = [\dots \varphi_i(x) \dots] \in \ell_2$$

- For positive definite $k(x, x')$,

$$k(x, x') = \langle \varphi(x), \varphi(x') \rangle_{\mathcal{F}}$$

- The reproducing property: $\forall f \in \mathcal{F}$,

$$f(x) = \langle f(\cdot), \varphi(x) \rangle_{\mathcal{F}}$$

The mean trick

- Given $\mathbf{P}$ a Borel probability measure on $\mathcal{X}$, define feature map $\mu_{\mathbf{P}} \in \mathcal{F}$

$$\mu_{\mathbf{P}} = [\dots \mathbf{E}_{\mathbf{P}} [\varphi_i(\mathsf{x})] \dots]$$

- For positive definite $k(x, x')$,

$$\mathbf{E}_{\mathbf{P},\mathbf{Q}} k(\mathsf{x}, \mathsf{y}) = \langle \mu_{\mathbf{P}}, \mu_{\mathbf{Q}} \rangle_{\mathcal{F}}$$

for $\mathsf{x} \sim \mathbf{P}$ and $\mathsf{y} \sim \mathbf{Q}$.

- The mean trick: (we call $\mu_{\mathbf{P}}$ a mean/distribution embedding)

$$\mathbf{E}_{\mathbf{P}}(f(\mathsf{x})) =: \langle \mu_{\mathbf{P}}, f(\cdot) \rangle_{\mathcal{F}}$$

MMD as distance between feature means

The maximum mean discrepancy is the distance between feature means:

$$MMD^2(\mathbf{P},\mathbf{Q}) = \|\mu_{\mathbf{P}} - \mu_{\mathbf{Q}}\|^2_{\mathcal{F}} = \langle \mu_{\mathbf{P}}, \mu_{\mathbf{P}} \rangle_{\mathcal{F}} + \langle \mu_{\mathbf{Q}}, \mu_{\mathbf{Q}} \rangle_{\mathcal{F}} - 2 \langle \mu_{\mathbf{P}}, \mu_{\mathbf{Q}} \rangle_{\mathcal{F}}$$
$$= \underbrace{\mathbf{E}_{\mathbf{P}} k(\mathsf{x},\mathsf{x}')}_{\text{(a)}} + \underbrace{\mathbf{E}_{\mathbf{Q}} k(\mathsf{y},\mathsf{y}')}_{\text{(a)}} - 2\underbrace{\mathbf{E}_{\mathbf{P},\mathbf{Q}} k(\mathsf{x},\mathsf{y})}_{\text{(b)}}$$

(a)= within distrib. similarity, (b)= cross-distrib. similarity

Proof:

$$\begin{aligned}
\|\mu_{\mathbf{P}} - \mu_{\mathbf{Q}}\|^2_{\mathcal{F}} &= \langle \mu_{\mathbf{P}} - \mu_{\mathbf{Q}}, \mu_{\mathbf{P}} - \mu_{\mathbf{Q}} \rangle_{\mathcal{F}} \\
&= \langle \mu_{\mathbf{P}}, \mu_{\mathbf{P}} \rangle + \langle \mu_{\mathbf{Q}}, \mu_{\mathbf{Q}} \rangle - 2 \langle \mu_{\mathbf{P}}, \mu_{\mathbf{Q}} \rangle \\
&= \mathbf{E}_{\mathbf{P}}[\mu_{\mathbf{P}}(\mathsf{x})] + \dots \\
&= \mathbf{E}_{\mathbf{P}} \langle \mu_{\mathbf{P}}(\cdot), k(\mathsf{x},\cdot) \rangle + \dots \\
&= \mathbf{E}_{\mathbf{P}} k(\mathsf{x},\mathsf{x}') + \mathbf{E}_{\mathbf{Q}} k(\mathsf{y},\mathsf{y}') - 2\mathbf{E}_{\mathbf{P},\mathbf{Q}} k(\mathsf{x},\mathsf{y})
\end{aligned}$$

Function F vs Feature Mean

- The (kernel) MMD: [ISMB06, NIPS06a]

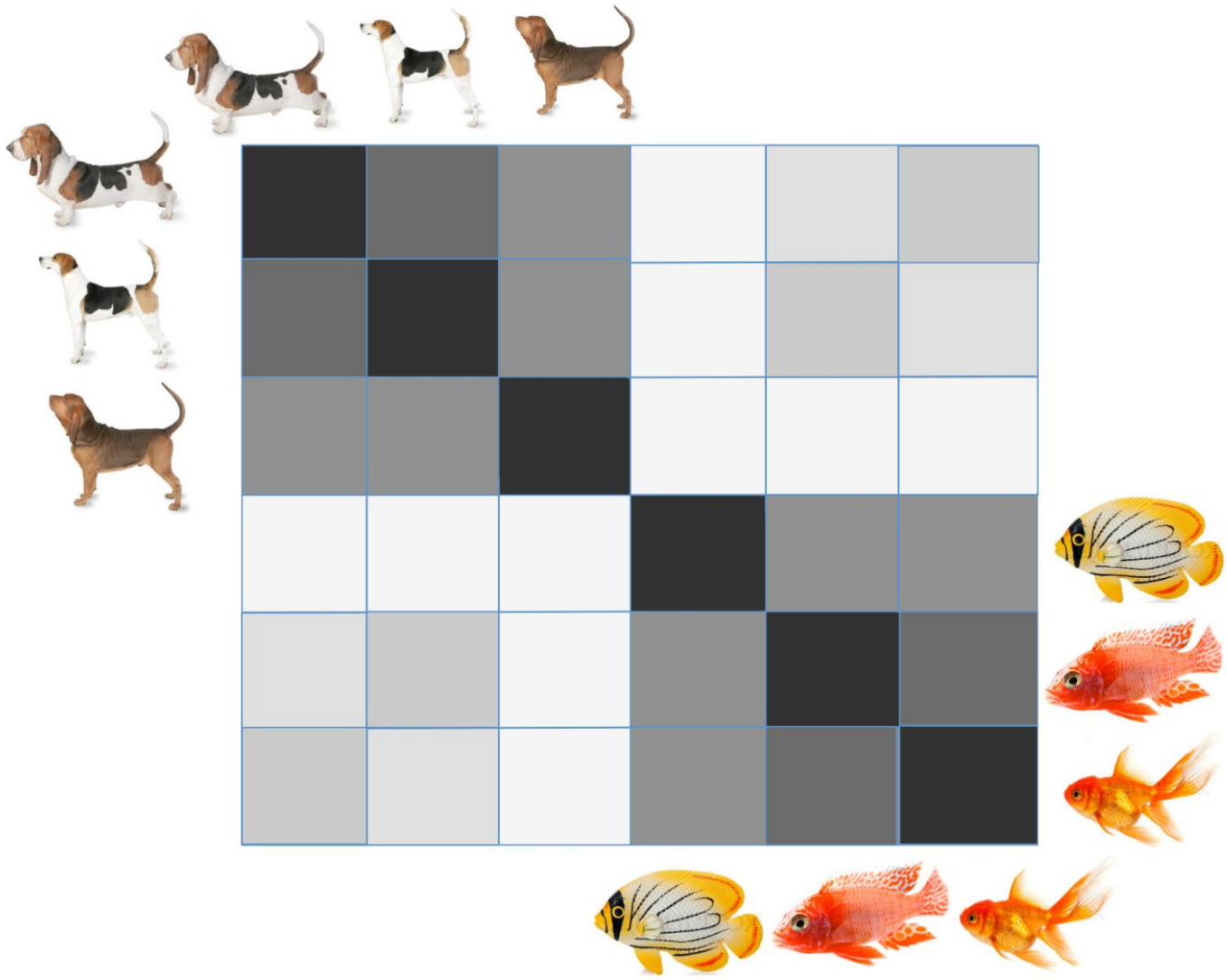
$$\begin{aligned}
&\mathrm{MMD}(\mathbf{P}, \mathbf{Q}; F) \\
&= \sup_{f \in F} \left[\mathbf{E}_{\mathbf{P}} f(\mathsf{x}) - \mathbf{E}_{\mathbf{Q}} f(\mathsf{y}) \right] \\
&= \sup_{f \in F} \left\langle f, \mu_{\mathbf{P}} - \mu_{\mathbf{Q}} \right\rangle_{\mathcal{F}} \\
&= \left\| \mu_{\mathbf{P}} - \mu_{\mathbf{Q}} \right\|_{\mathcal{F}}
\end{aligned}$$

use

$$\|\theta\|_{\mathcal{F}} = \sup_{f \in F} \langle f, \theta \rangle_{\mathcal{F}}$$

since $F := \{ f \in \mathcal{F} : \|f\| \leq 1 \}$

Function view and feature view equivalent



PhEMD

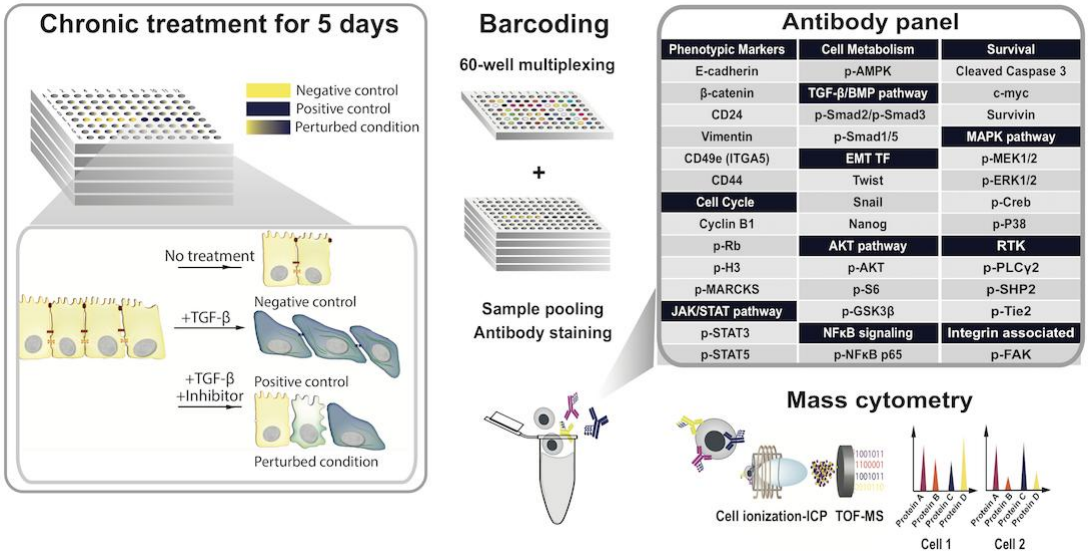
Phenotypic EMD (Chen et al. Nature Methods 2020)

William Chen

Main idea: **Derive an embedding of multi-sample datasets on the basis of earth mover's distance**

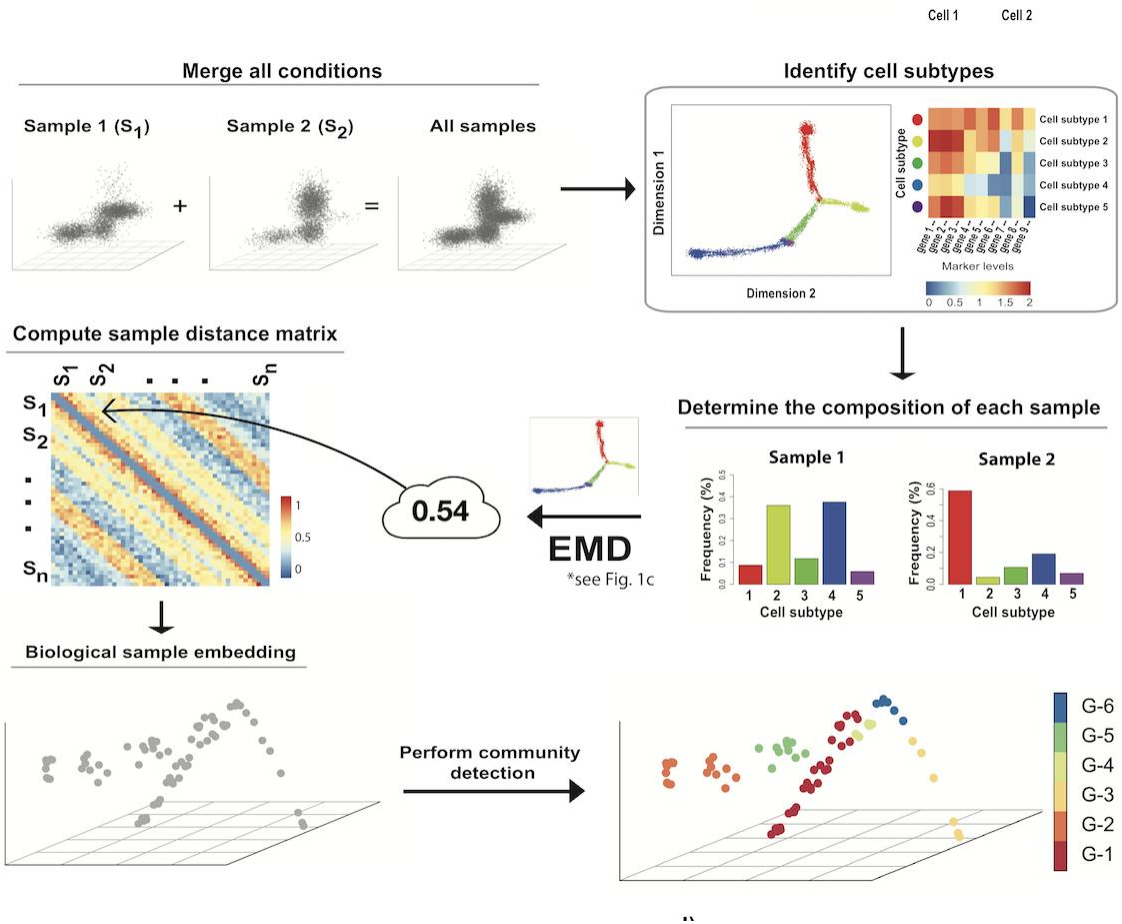
Will Chen
Nevana Zivanovic
David van Dijk
Guy Wolf
Bernd Bodenmiller

Drug Perturbation Data

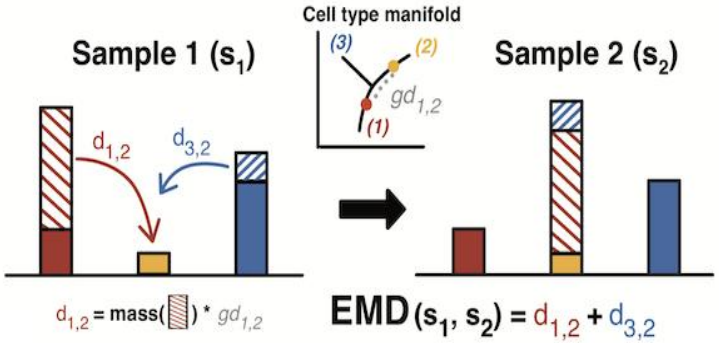


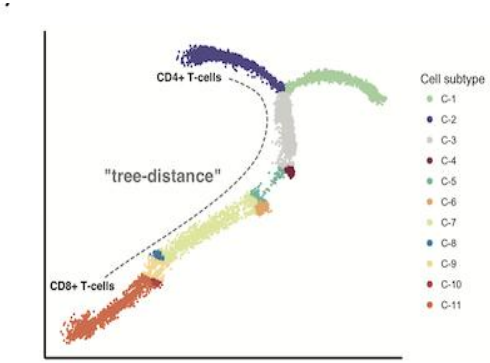
HMLE Cells Induced with TGFB and perturbed by 300 drugs

Overview


Cell 1
Cell 2
Merge all conditions
Sample 1 (S_1)
Sample 2 (S_2)
All samples
Identify cell subtypes
Dimension 1
Dimension 2
Cell subtype
Cell subtype 1
Cell subtype 2
Cell subtype 3
Cell subtype 4
Cell subtype 5
Marker levels
0 0.5 1 1.5 2
Compute sample distance matrix
S_1 S_2 . . . S_n
1
0.5
0
0.54
EMD
*see Fig. 1c
Determine the composition of each sample
Sample 1
Sample 2
Frequency (%)
Cell subtype
Biological sample embedding
Perform community detection
G-6
G-5
G-4
G-3
G-2
G-1

Ground Distance





Drug Perturbation Landscape