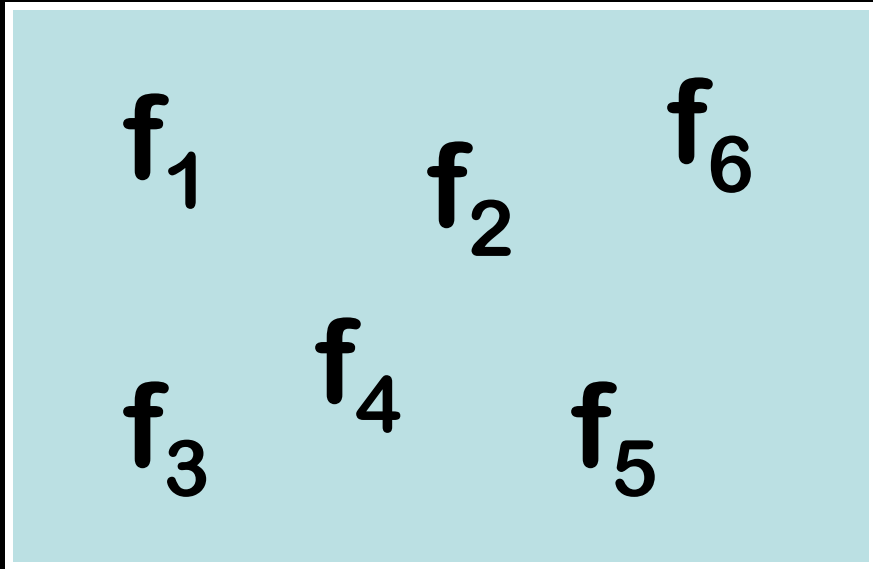


# Density estimation in linear time (+approximating $L_1$ -distances)

Satyaki Mahalanabis  
Daniel Štefankovič

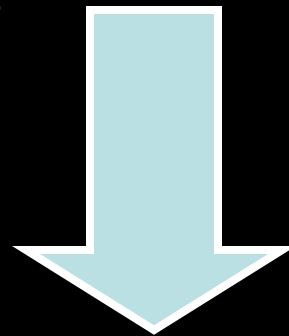
University of Rochester

# Density estimation



**+ DATA**

**F = a family of densities**



**density**

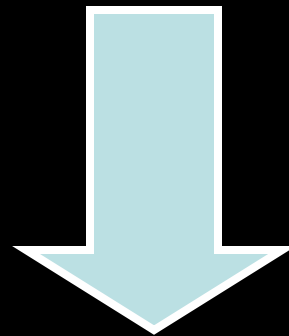
# Density estimation - example

$$N(\mu, 1)$$

+

0.418974,  
0.848565,  
1.73705,  
1.59579,  
-1.18767,  
-1.05573,  
-1.36625

F = a family of normal  
densities with  $\sigma=1$



$\mu$

# Measure of quality:

$g$ =TRUTH

$f$ =OUTPUT

$L_1$  – distance from the truth

$$\|f-g\|_1 = \int |f(x)-g(x)| dx$$

Why  $L_1$ ?

- 1) small  $L_1 \Rightarrow$  all events estimated with small additive error
- 2) scale invariant

# Obstacles to “quality”:

**F**

**+ DATA**

**bad data**

**weak class  
of densities**

**$\text{dist}_1(g, F)$**

**$\Delta ?$**

# What is bad data ? ~~$|h-g|_1$~~

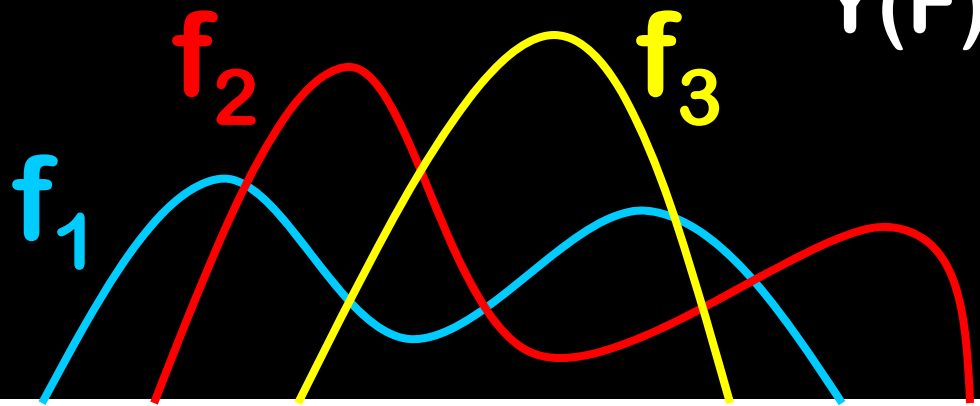
$g = \text{TRUTH}$

$h = \text{DATA}$  (empirical density)

$$\Delta = 2 \max_{A \in Y(F)} |h(A) - g(A)|$$

$Y(F) = \text{Yatracos class of } F$

$$A_{ij} = \{ x \mid f_i(x) > f_j(x) \}$$



$A_{12}$

$A_{13}$

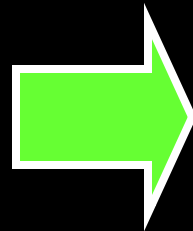
$A_{23}$

# Density estimation

**F**

+

**DATA (h)**



**f** with small  $|g-f|_1$

assuming these are small:

$\text{dist}_1(g, F)$

$$\Delta = 2 \max_{A \in Y(F)} |h(A) - g(A)|$$

# Why would these be small ???

$\text{dist}_1(h, F)$

$$\Delta = 2 \max_{A \in Y(F)} |h(A) - g(A)|$$

## They will be if:

1) pick a large enough  $F$

2) pick a small enough  $F$

so that VC-dimension of  $Y(F)$  is small

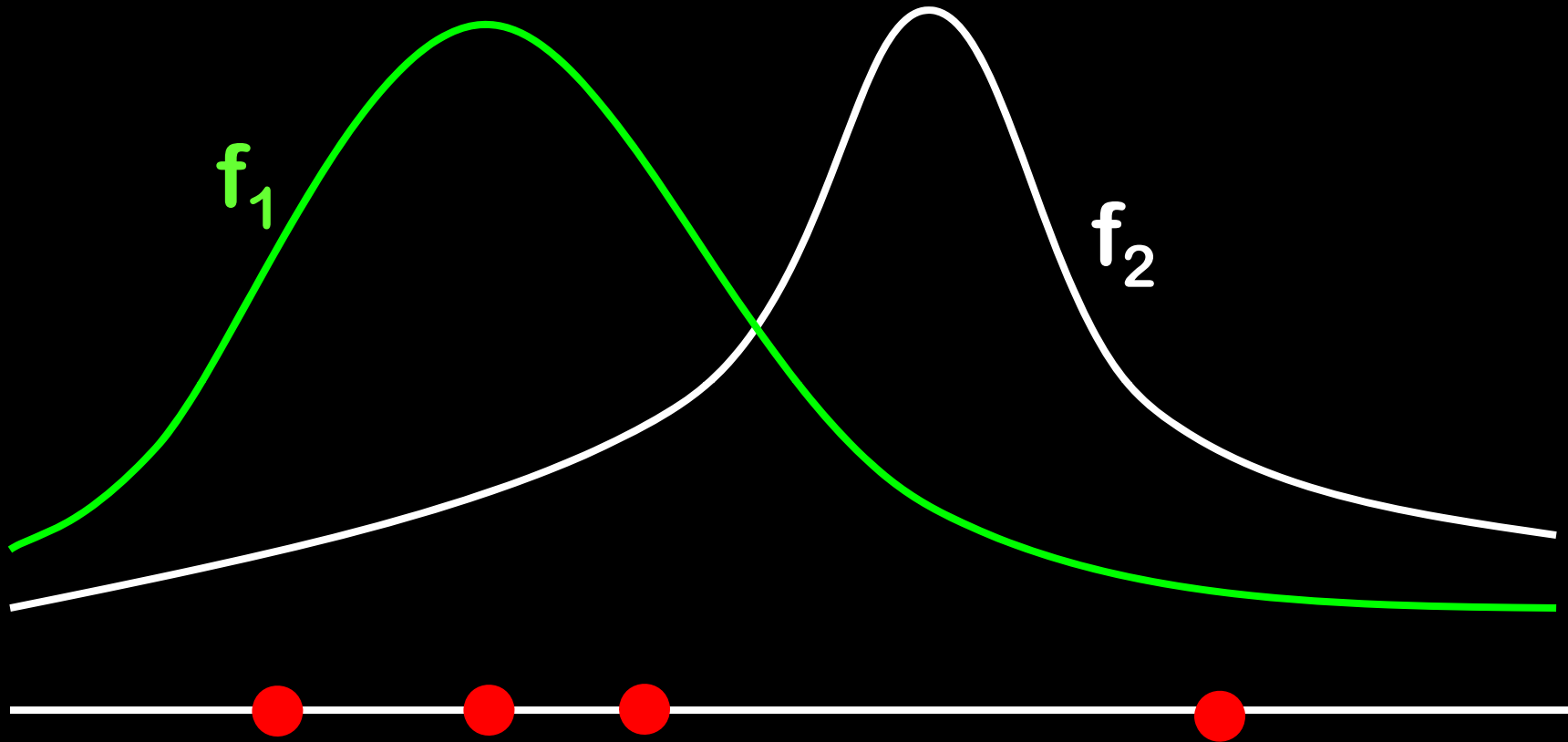
3) data are iid from  $h$

Theorem (Haussler, Dudley, Vapnik, Chervonenkis):

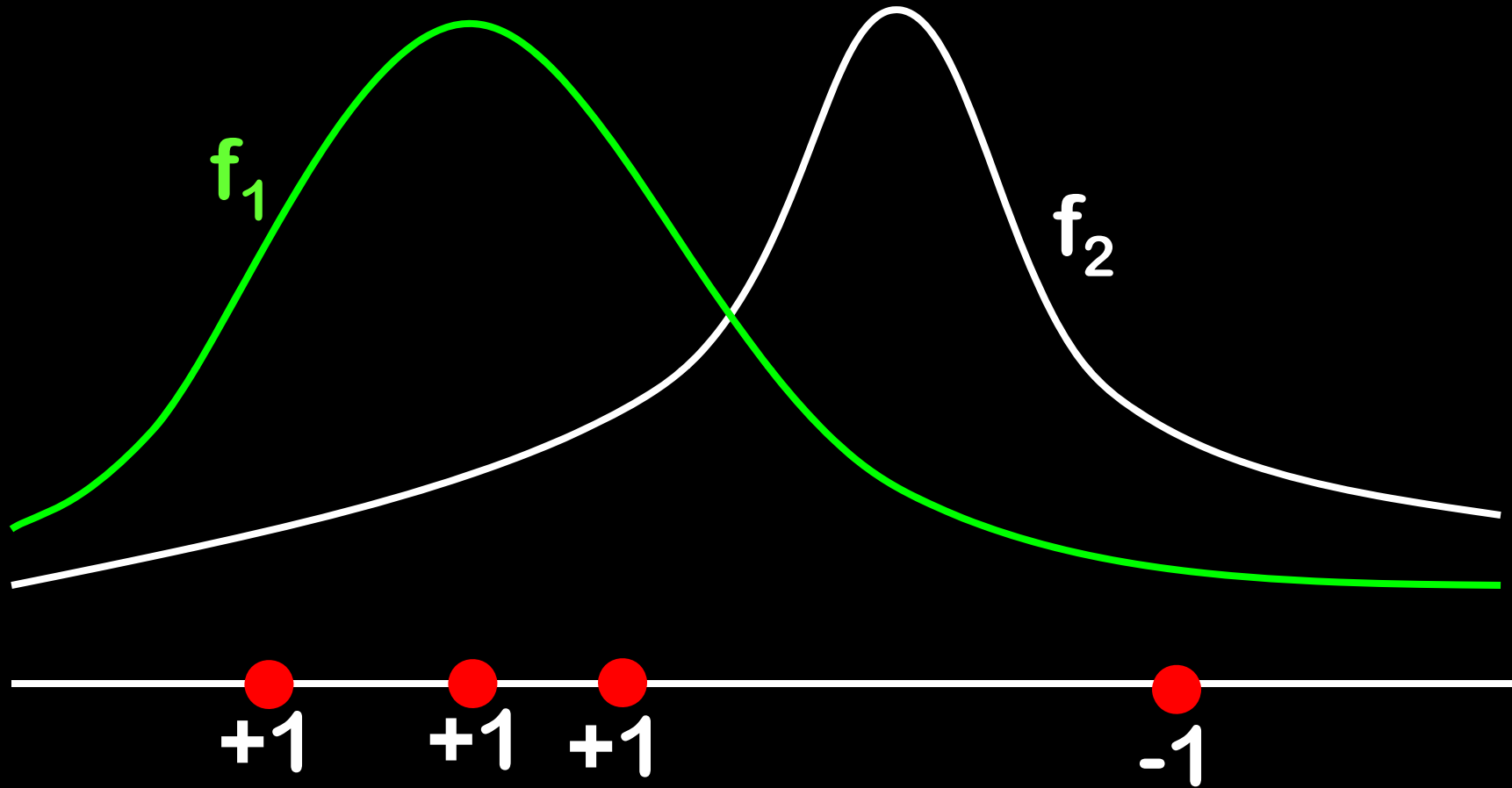
$$E[\max_{A \in Y} |h(A) - g(A)|] \leq \sqrt{\frac{\text{VC}(Y)}{\text{samples}}}$$



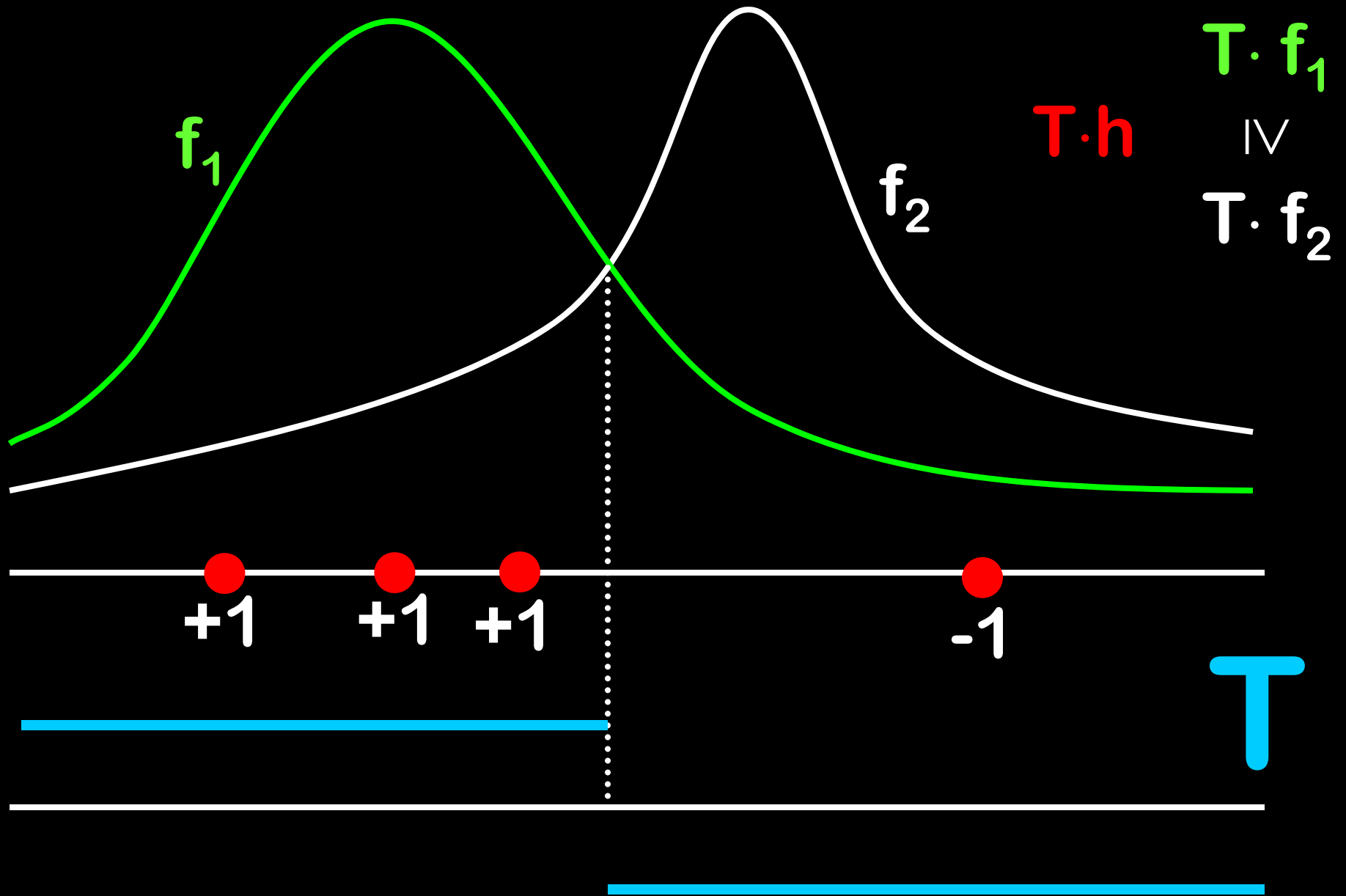
# How to choose from 2 densities?



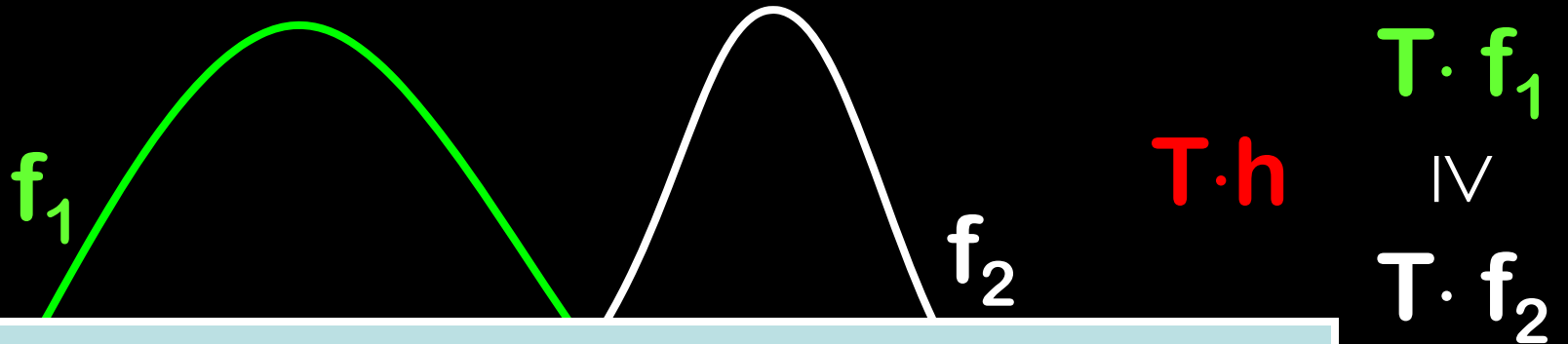
# How to choose from 2 densities?



# How to choose from 2 densities?



# How to choose from 2 densities?



**Scheffé:**

if  $T \cdot h > T \cdot (f_1 + f_2) / 2 \Rightarrow f_1$   
else  $\Rightarrow f_2$

**Theorem (see DL'01):**

$$|f - g|_1 \leq 3 \text{dist}_1(g, F) + 2\Delta$$

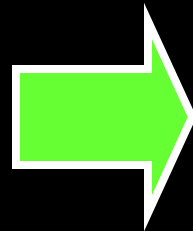
**T**

# Density estimation

**F**

+

**DATA (h)**



**f** with small  $|g-f|_1$

assuming these are small:

$\text{dist}_1(g, F)$

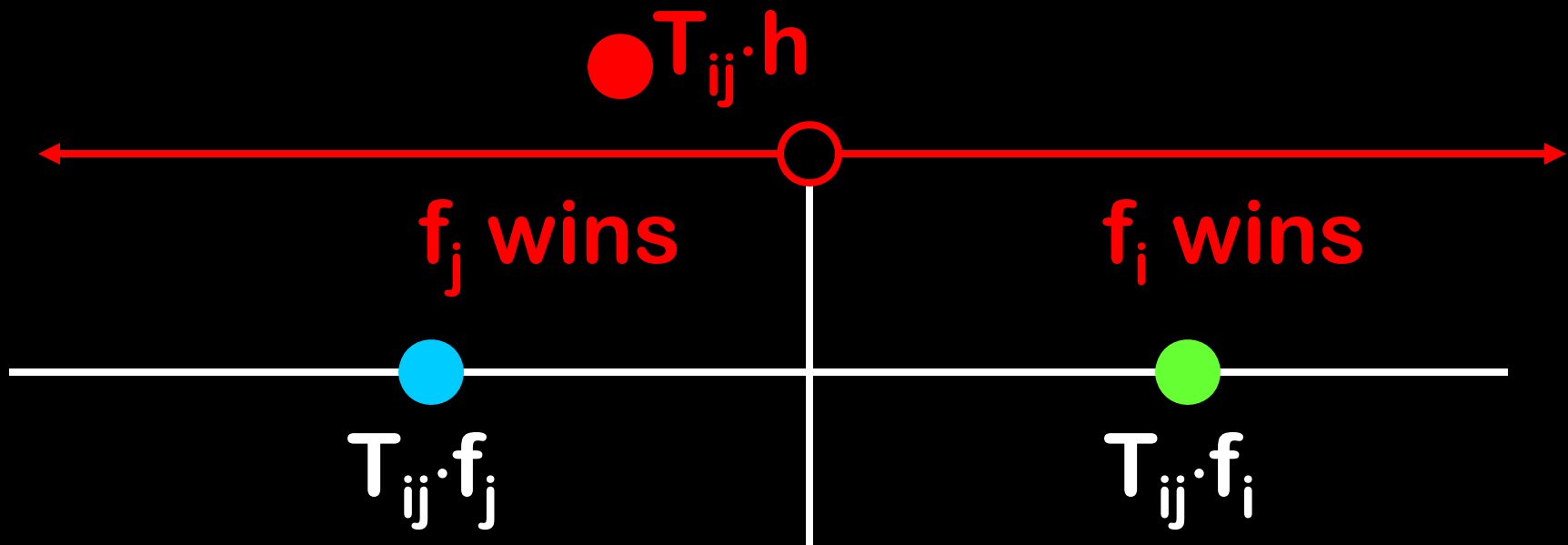
$$\Delta = 2 \max_{A \in Y(F)} |h(A) - g(A)|$$

# Test functions

$$F = \{f_1, f_2, \dots, f_N\}$$

$$T_{ij}(x) = \text{sgn}(f_i(x) - f_j(x))$$

$$T_{ij} \cdot (f_i - f_j) = \int (f_i - f_j) \text{sgn}(f_i - f_j) = |f_i - f_j|_1$$



# Density estimation algorithms

## Scheffé tournament:

Pick the density with the most wins.

## Theorem (DL'01):

$$|f-g|_1 \leq 9 \text{dist}_1(g, F) + 8\Delta$$

$n^2$

## Minimum distance estimate (Y'85):

Output  $f_k \in F$  that minimizes

$$\max_{ij} |(f_k - h) \cdot T_{ij}|$$

$n^3$

## Theorem (DL'01):

$$|f-g|_1 \leq 3 \text{dist}_1(g, F) + 2\Delta$$

# Density estimation algorithms

**Scheffé tournament:**

Pick the density with the most wins.

**Theorem (DL'01):**

$$|f-g|_1 \leq 9 \text{dist}_1(g, F) + 8\Delta$$

$n^2$

Can we do better?

(Y'85):

es

$n^3$

**Theorem (DL'01):**

$$|f-g|_1 \leq 3 \text{dist}_1(g, F) + 2\Delta$$



# Our algorithm:

## Efficient minimum loss-weight

repeat until one distribution left

- 1) pick the pair of distributions in  $F$  that are furthest apart (in  $L_1$ )
- 2) eliminate the loser

### Theorem [MS'08]:

$$|f-g|_1 \leq 3 \text{dist}_1(g, F) + 2\Delta$$

$n^*$

Take the most “discriminative” action.

# Tournament revelation problem

## INPUT:

a weighed undirected graph  $G$   
(wlog all edge-weights distinct)

## OUTPUT:

REPORT: heaviest edge  $\{u_1, v_1\}$  in  $G$

**ADVERSARY eliminates  $u_1$  or  $v_1 \mapsto G_1$**

REPORT: heaviest edge  $\{u_2, v_2\}$  in  $G_1$

**ADVERSARY eliminates  $u_2$  or  $v_2 \mapsto G_2$**

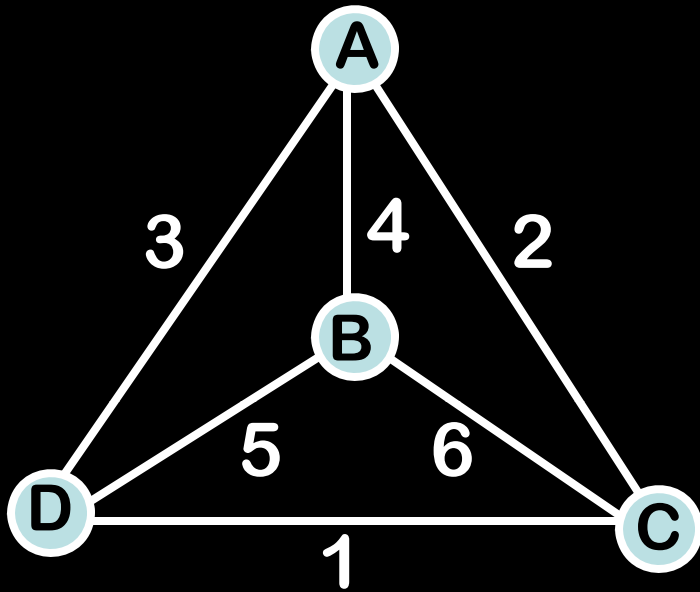
.....

## OBJECTIVE:

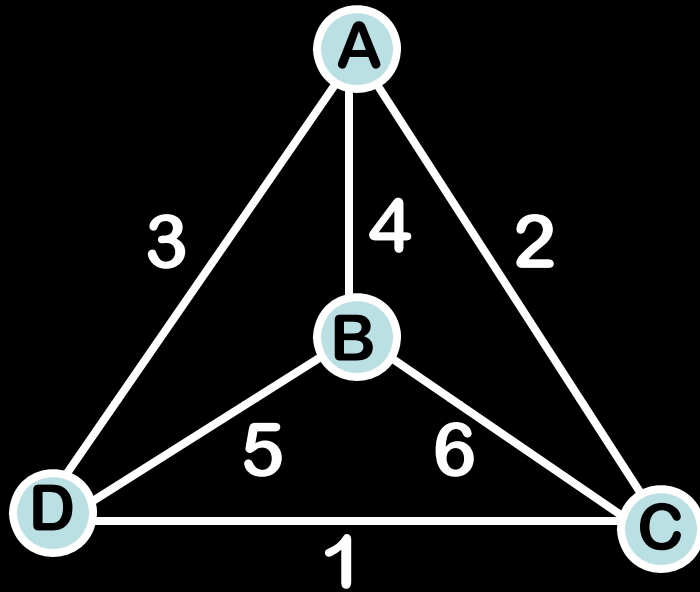
minimize total time spent generating reports

# Tournament revelation problem

report the heaviest edge



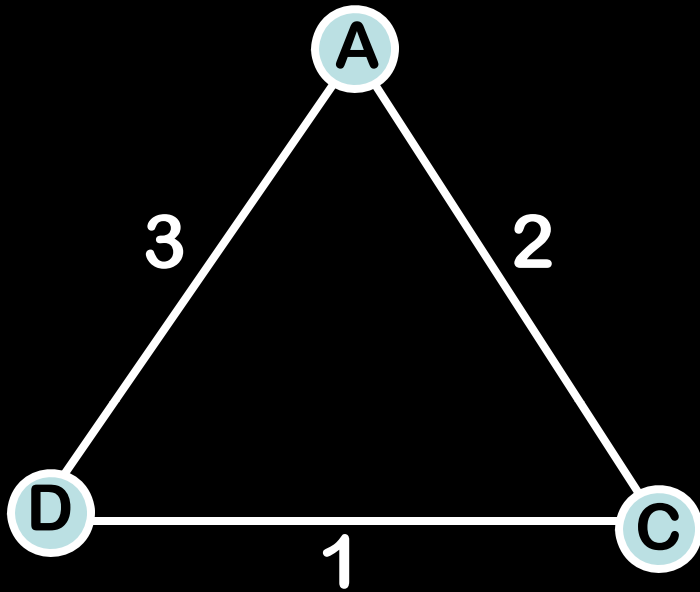
# Tournament revelation problem



report the heaviest edge

**BC**

# Tournament revelation problem



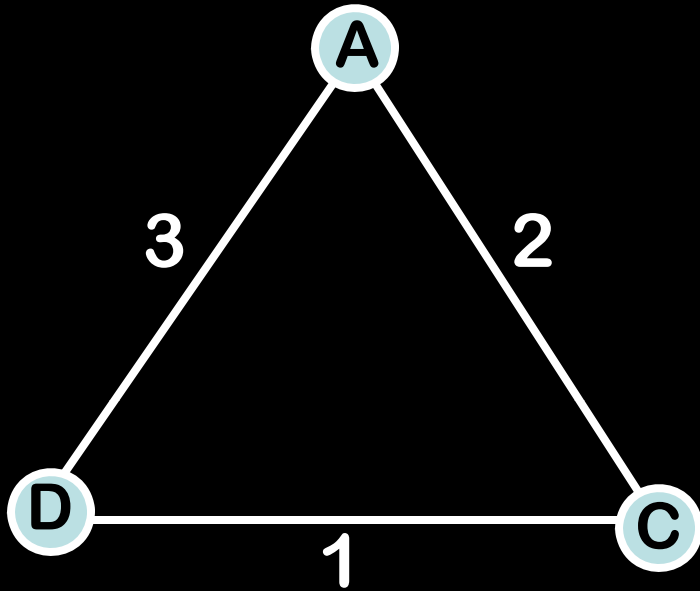
report the heaviest edge

**BC**

**eliminate B**

report the heaviest edge

# Tournament revelation problem



report the heaviest edge

**BC**

**eliminate B**

report the heaviest edge

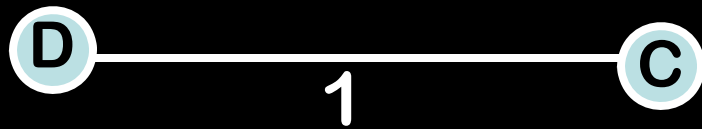
**AD**

# Tournament revelation problem

report the heaviest edge

**BC**

eliminate **B**



report the heaviest edge

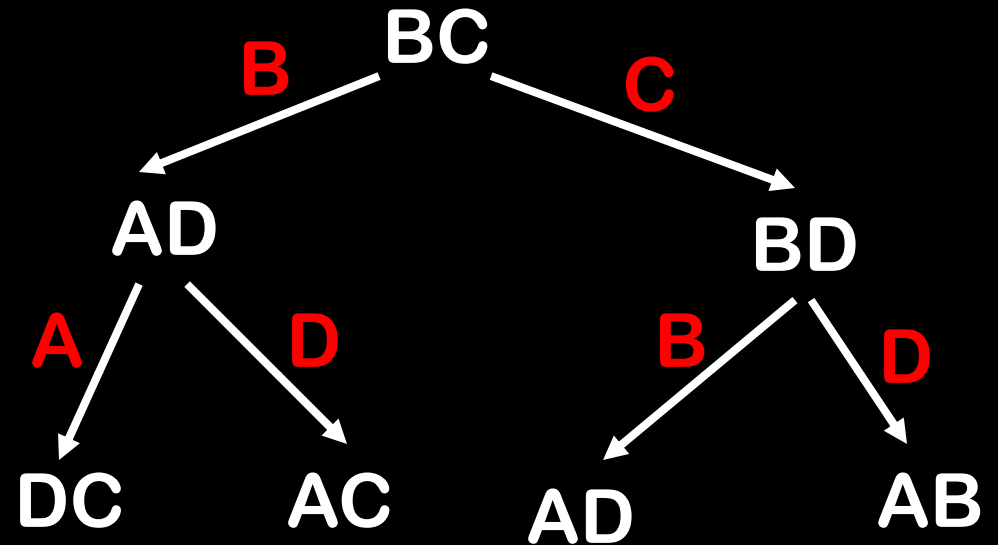
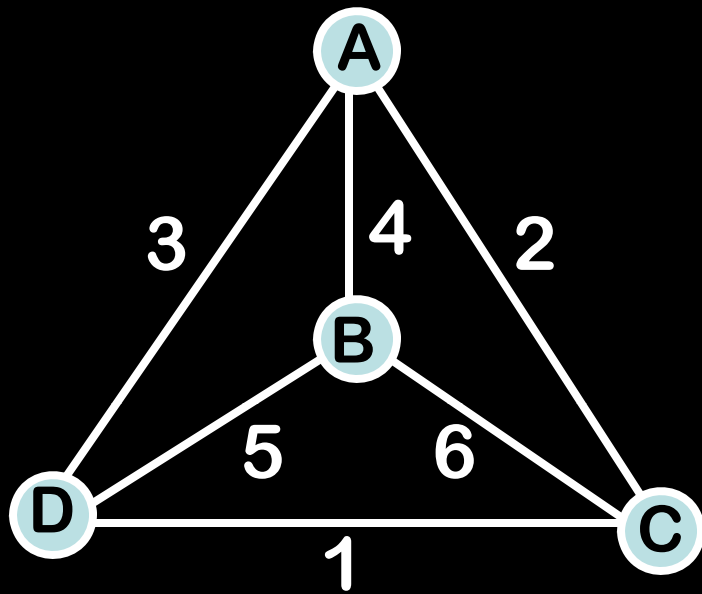
**AD**

eliminate **A**

report the heaviest edge

**CD**

# Tournament revelation problem



$2^{O(F)}$  preprocessing  $\Rightarrow O(F)$  run-time

$O(F^2 \log F)$  preprocessing  $\Rightarrow O(F^2)$  run-time

**WE DO NOT KNOW:**

Can get  $O(F)$  run-time with  
polynomial preprocessing ???



# Efficient minimum loss-weight

repeat until one distribution left

- 1) pick the pair of distributions that are furthest apart (in  $L_1$ )
- 2) eliminate the loser

(in practice 2) is more costly)

$2^{O(F)}$  preprocessing  $\Rightarrow O(F)$  run-time

$O(F^2 \log F)$  preprocessing  $\Rightarrow O(F^2)$  run-time

**WE DO NOT KNOW:**

Can get  $O(F)$  run-time with polynomial preprocessing ???

# Efficient minimum loss-weight

repeat until one distribution left

- 1) pick the pair of distributions that are furthest apart (in  $L_1$ )
- 2) eliminate the loser

**Theorem:**

$$|f-g|_1 \leq 3 \text{dist}_1(g, F) + 2\Delta$$

**n**

**Proof:**

“that guy lost even more badly!”

For every  $f'$  to which  $f$  loses

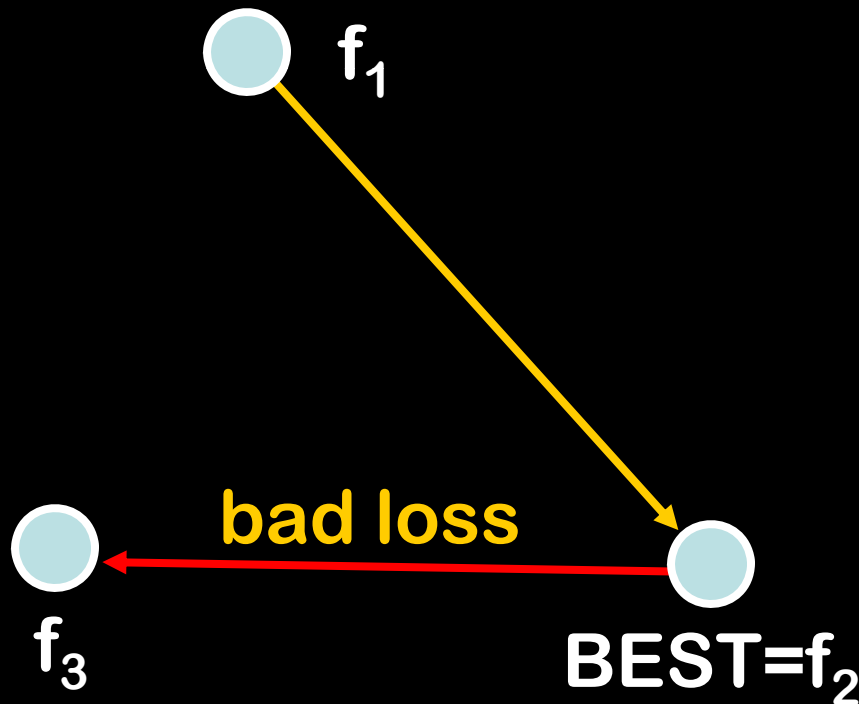
$$|f-f'|_1 \leq \max_{f' \text{ loses to } f''} |f'-f''|_1$$

**Proof:**

“that guy lost even more badly!”

For every  $f'$  to which  $f$  loses

$$|f-f'|_1 \leq \max_{f' \text{ loses to } f} |f'-f''|_1$$



$$2h \cdot T_{23} \leq f_2 \cdot T_{23} + f_3 \cdot T_{23}$$

$$(f_1 - f_2) \cdot T_{12} \leq (f_2 - f_3) \cdot T_{23}$$

$$(f_4 - h) \cdot T_{23} \leq \Delta$$

$$(f_i - f_j) \cdot (T_{ii} - T_{kl}) \geq 0$$

---

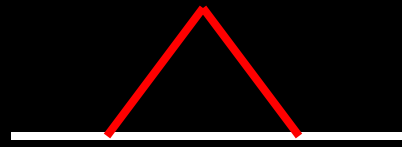
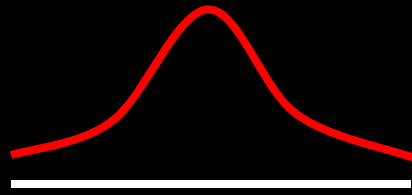
$$|f_1 - g|_1 \leq 3|f_2 - g|_1 + 2\Delta$$

# Application:

## kernel density estimates

(Akaike'54, Parzen'62, Rosenblatt'56)

**K** = kernel



**h** = density

kernel used to smooth empirical **g**  
( $x_1, x_2, \dots, x_n$  i.i.d. samples from **h**)

$$\frac{1}{n} \sum_{i=1}^n K(y-x_i) \xrightarrow{\text{as } n \rightarrow \infty} h * K$$

=

$$g * K$$

# What K should we choose?

$$\frac{1}{n} \sum_{i=1}^n K(y-x_i) \stackrel{g^* K}{=} \xrightarrow{\text{as } n \rightarrow \infty} h^* K$$

Dirac  $\delta$  is not good

Dirac  $\delta$  would be good

Something in-between: **bandwidth selection**  
for kernel density estimates

$$K_s(x) = \frac{K(x/s)}{s}$$

as  $s \rightarrow 0$

$K_s(x) \rightarrow$  Dirac  $\delta$

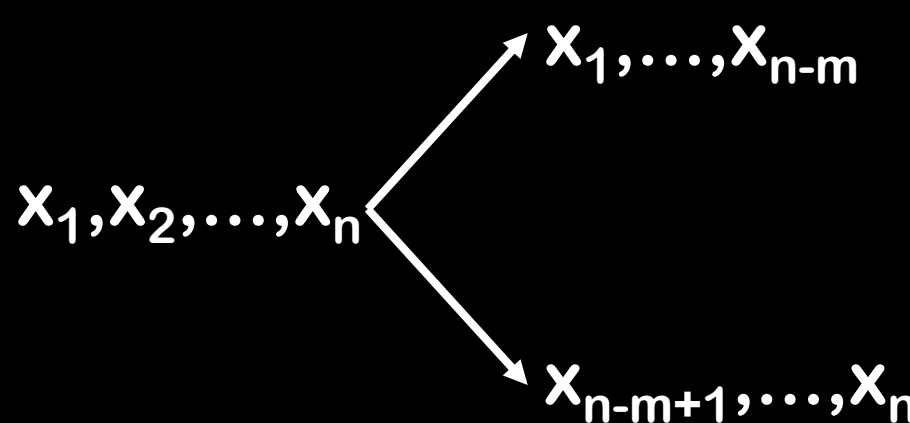
**Theorem (see DL'01):** as  $s \rightarrow 0$  with  $sn \rightarrow \infty$

$$|g^* K - h|_1 \rightarrow 0$$

# Data splitting methods for kernel density estimates

How to pick the **smoothing factor** ?

$$\frac{1}{ns} \sum_{i=1}^n k\left(\frac{y-x_i}{s}\right)$$



The diagram shows a set of data points  $x_1, x_2, \dots, x_n$  on the left. Two arrows branch out from this set: one pointing up and to the right to the subset  $x_1, \dots, x_{n-m}$ , and another pointing down and to the right to the subset  $x_{n-m+1}, \dots, x_n$ .

$$f_s = \frac{1}{(n-m)s} \sum_{i=1}^{n-m} k\left(\frac{y-x_i}{s}\right)$$

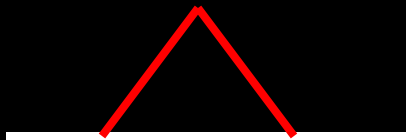
choose **s** using  
density estimation

# Kernels we will use:

$$\frac{1}{ns} \sum k\left(\frac{y-x_i}{s}\right)$$



piecewise uniform



piecewise linear

# Bandwidth selection for uniform kernels

E.g.  $N \approx n^{1/2}$   
 $m \approx n^{5/4}$

$N$  distributions

each is piecewise uniform with  $n$  pieces

$m$  datapoints

Goal: run the density estimation algorithm efficiently

	TIME	MD	EMLW
$g \cdot T_{ij} \geq \frac{(f_i + f_j) \cdot T_{ij}}{2}$	$n + m \log n$		$N$
$(f_k - h) \cdot T_{kj}$	$n + m \log n$	$N^2$	
$ f_i - f_j _1$	$n$		$N^2$



# Bandwidth selection for uniform kernels

**N** distributions  
 each is piecewise  
**m** datapoints

Can speed this up?

$$N \approx n^{1/2}$$

$$m \approx n^{5/4}$$

pieces

Goal: run the density estimation algorithm efficiently

	TIME	MD	EMLW
$g \cdot T_{ij} \geq \frac{(f_i + f_j) \cdot T_{ij}}{2}$	$n + m \log n$		$N$
$(f_k - h) \cdot T_{kj}$	$n + m \log n$	$N^2$	
$ f_i - f_j _1$	$n$		$N^2$

# Bandwidth selection for uniform kernels

**N** distributions  
 each is piecewise  
**m** datapoints

Goal: run the density

$$N \approx n^{1/2}$$

$$m \approx n^{5/4}$$

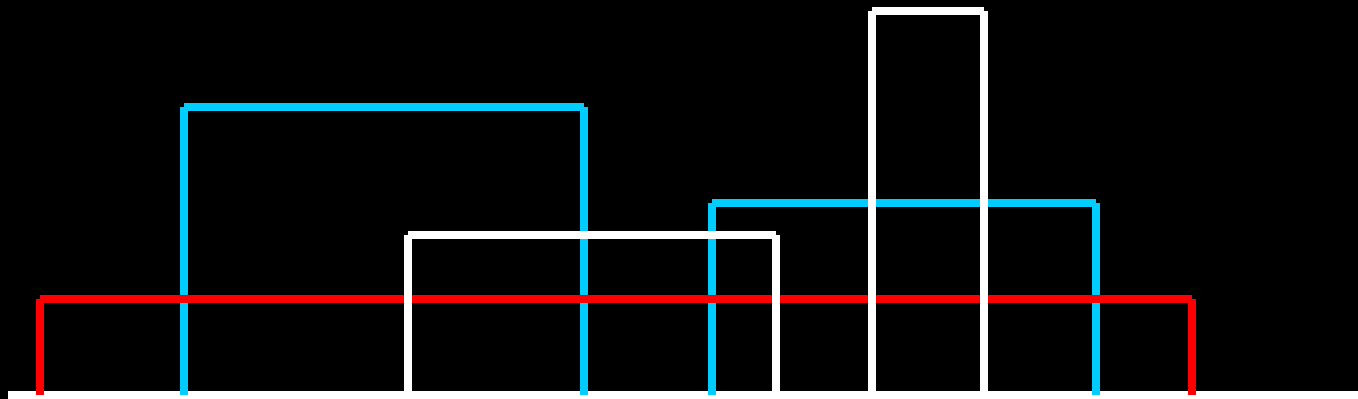
Can speed  
 this up?

absolute error bad  
 relative error good

	TIME	MEM	ERROR
$g \cdot T_{ij} \geq \frac{(f_i + f_j) \cdot T_{ij}}{2}$	$n + m \log n$		$N$
$(f_k - h) \cdot T_{kj}$	$n + m \log n$	$N^2$	
$ f_i - f_j _1$	$n$		$N^2$

# Approximating $L_1$ -distances between distributions

$N$  piecewise uniform densities (each  $n$  pieces)



**WE WILL DO:**  $\frac{(N^2 + Nn) (\log N)}{\epsilon^2}$

**TRIVIAL (exact):**  $N^2 n$

# Dimension reduction for $L_2$

Johnson-Lindenstrauss Lemma ('82)

$|S|=n$

$$\phi: L_2 \rightarrow L_2^t \quad t = O(\varepsilon^{-2} \ln n)$$

$(\forall x, y \in S)$

$$d(x, y) \leq d(\phi(x), \phi(y)) \leq (1 + \varepsilon)d(x, y)$$

$$N(0, t^{-1/2})$$

# Dimension reduction for $L_1$

Cauchy Random Projection (Indyk'00)  $|S|=n$

$$\phi: L_1 \rightarrow L_1^t \quad t = O(\varepsilon^{-2} \ln n)$$

$(\forall x, y \in S)$

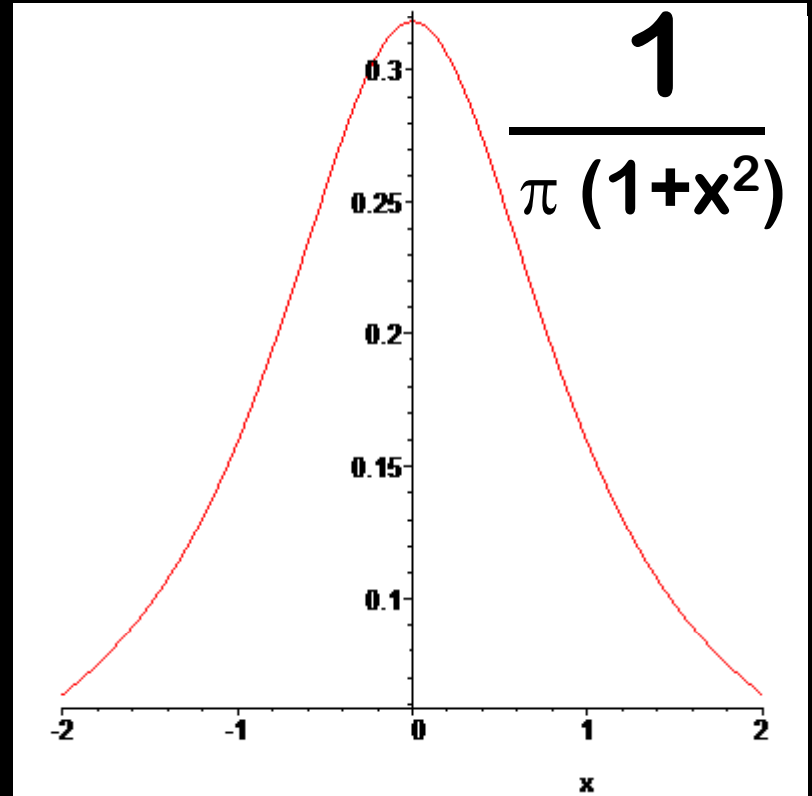
$$d(x, y) \leq \mathbf{est}(\phi(x), \phi(y)) \leq (1 + \varepsilon)d(x, y)$$

$$C(0, 1/t)$$

(Charikar, Brinkman'03 : cannot replace est by d)

# Cauchy distribution $C(0,1)$

density function:



## FACTS:

$$X \sim C(0,1)$$

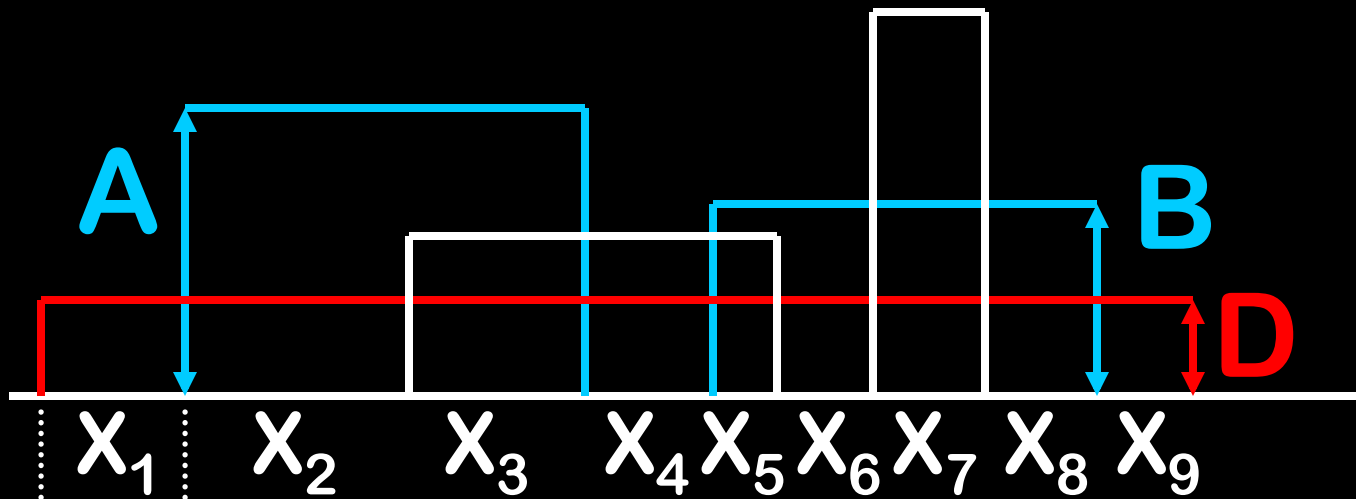
$$\Rightarrow aX \sim C(0,|a|)$$

$$X \sim C(0,a), Y \sim C(0,b)$$

$$\Rightarrow X+Y \sim C(0,a+b)$$

# Cauchy random projection for $L_1$

(Indyk'00)



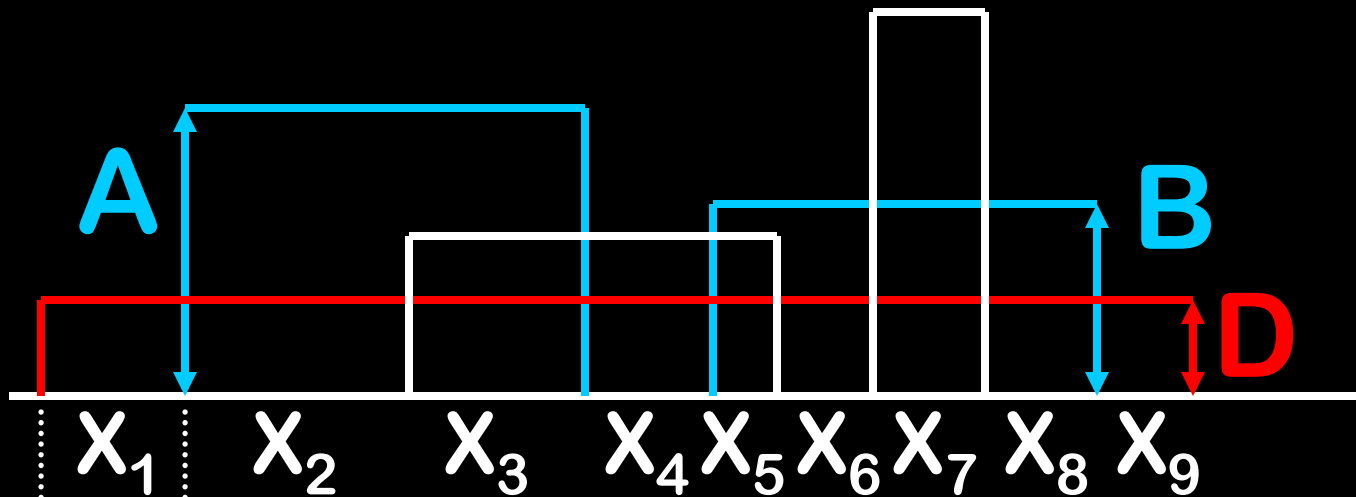
$$x_1 \sim C(0, z)$$

$$A(x_2 + x_3) + B(x_5 + x_6 + x_7 + x_8)$$

$z$

# Cauchy random projection for $L_1$

(Indyk'00)



$$X_1 \sim C(0, z)$$

$$A(X_2 + X_3) + B(X_5 + X_6 + X_7 + X_8)$$

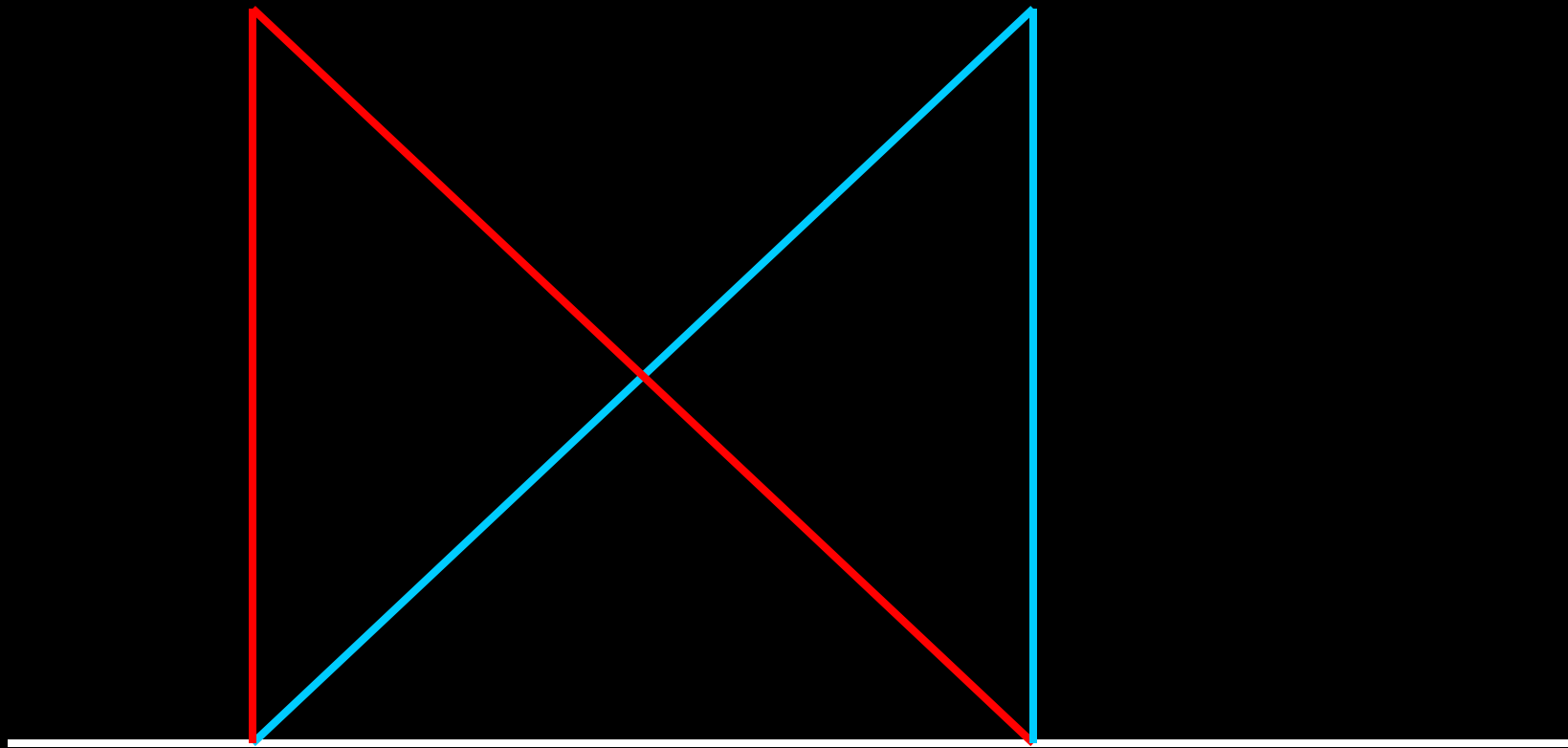
$$D(X_1 + X_2 + \dots + X_8 + X_9)$$

$$\sim \text{Cauchy}(0, |\square - \square|_1)$$



All pairs  $L_1$ -distances

piece-wise linear densities

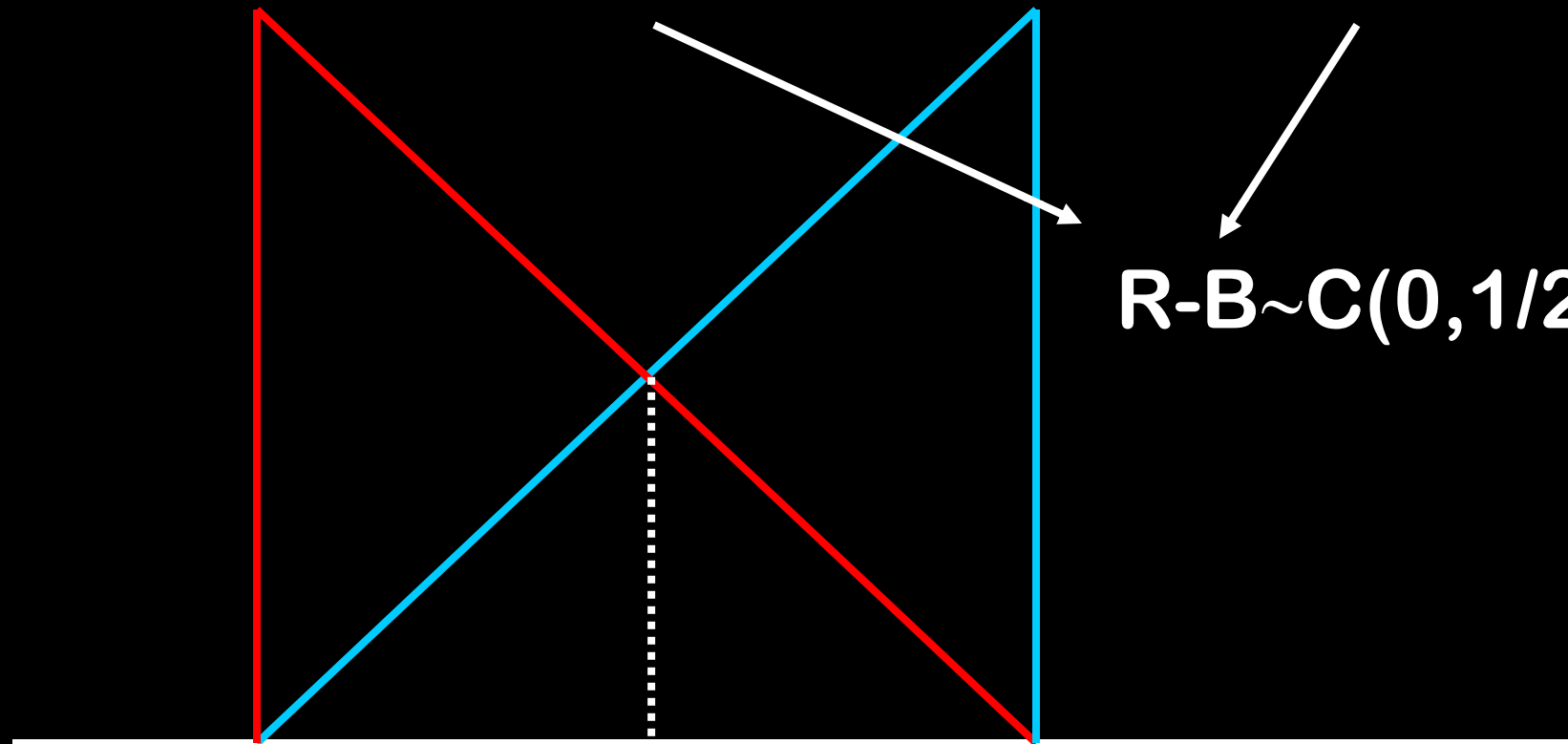


# All pairs $L_1$ -distances

piece-wise linear densities

$$R = (3/4)X_1 + (1/4)X_2$$

$$B = (3/4)X_2 + (1/4)X_1$$



$$R - B \sim C(0, 1/2)$$

$X_1$

$X_2$

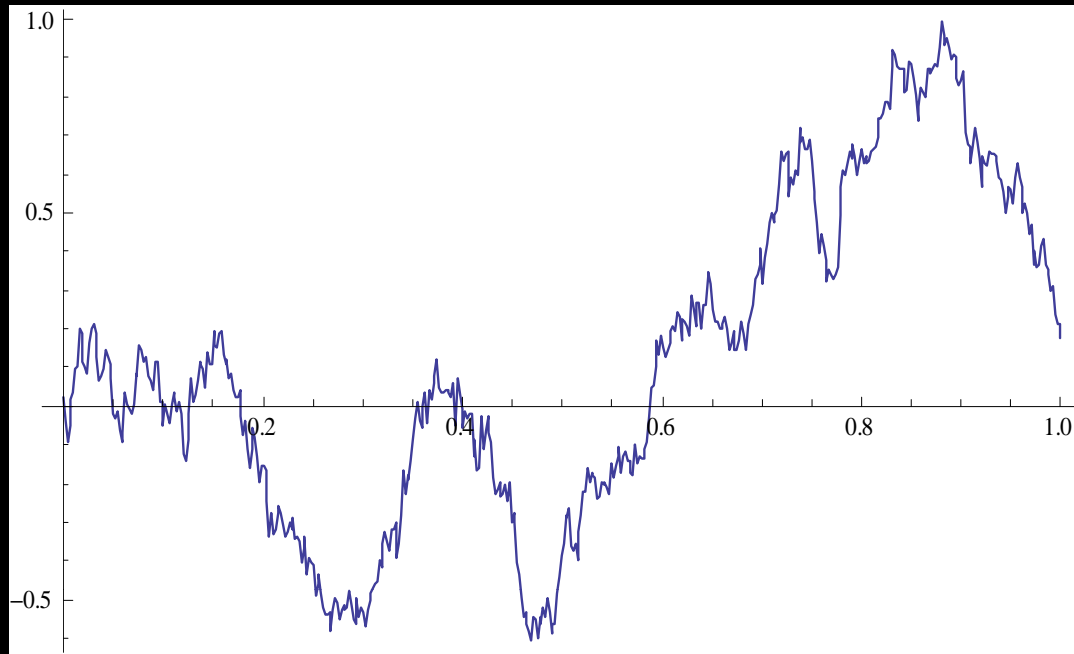
$\sim C(0, 1/2)$

**All pairs  $L_1$ -distances**

piece-wise linear densities

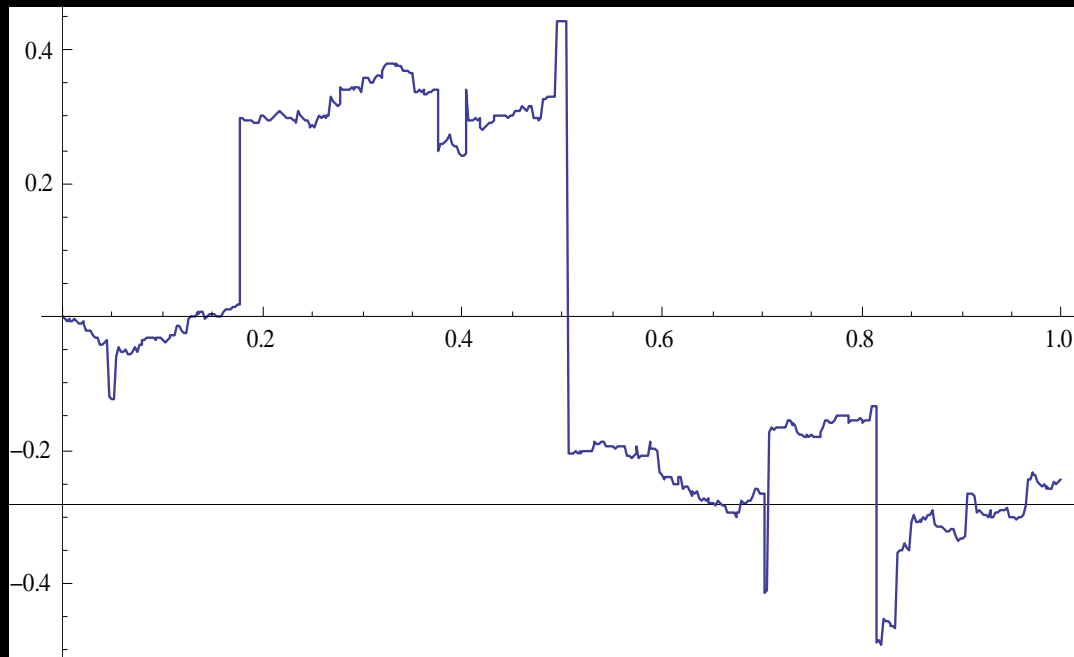
**Problem: too many intersections!**

**Solution: cut into even smaller pieces!**



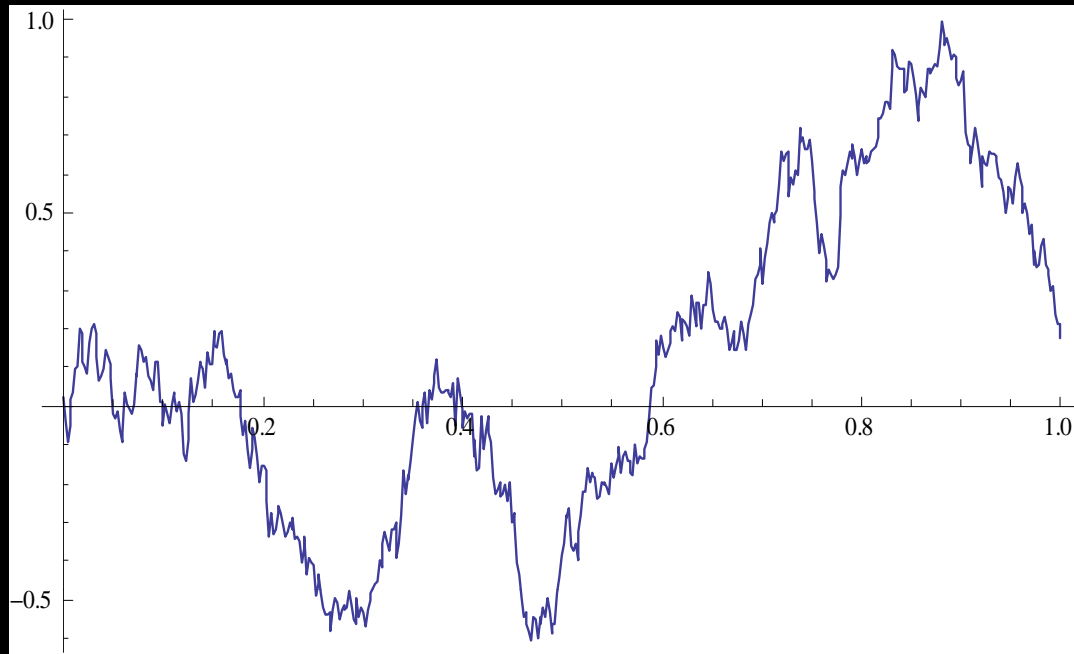
**Brownian motion**

$$\frac{1}{(2\pi)^{1/2}} \exp(-x^2/2)$$



**Cauchy motion**

$$\frac{1}{\pi (1+x)^2}$$



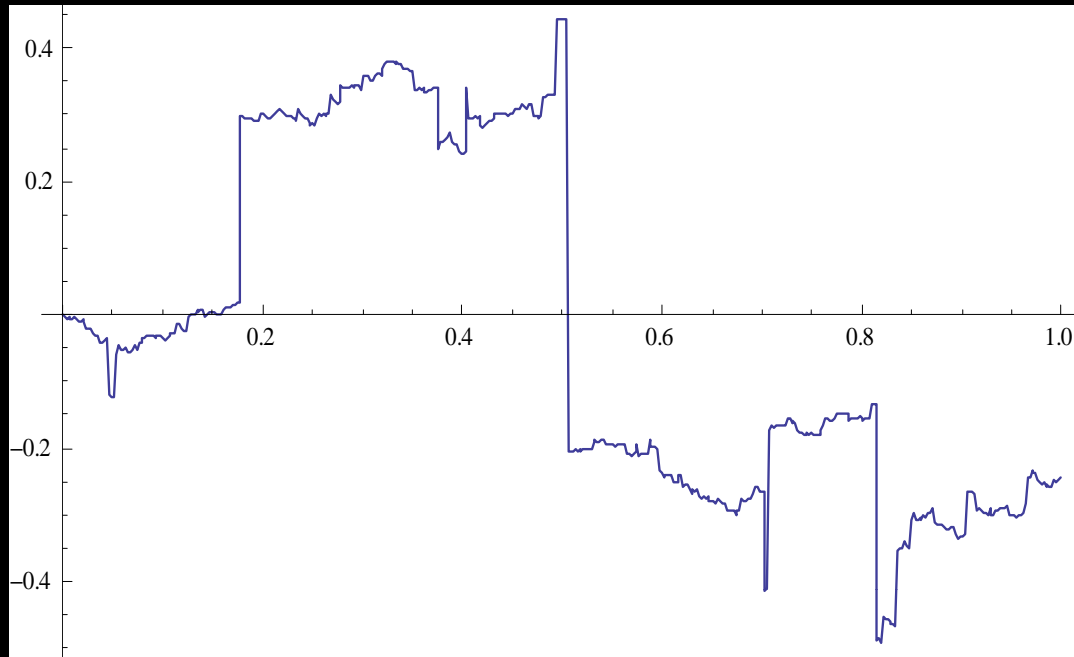
**Brownian motion**

$$\frac{1}{(2\pi)^{1/2}} \exp(-x^2/2)$$

**computing integrals is easy**

$$f: \mathbb{R} \rightarrow \mathbb{R}^d$$

$$\int f \, dL = Y \sim N(0, S)$$



Cauchy motion

$$\frac{1}{\pi (1+x)^2}$$

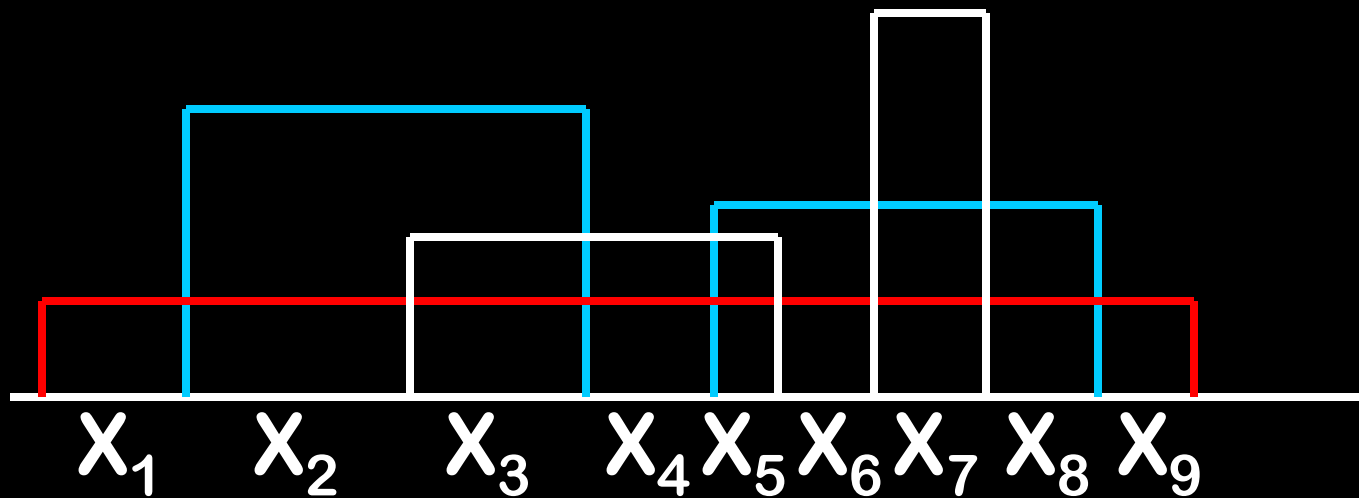
computing integrals is easy

$$f: \mathbb{R} \rightarrow \mathbb{R}^d$$

$$\int f \, dL = Y \sim C(0, s) \text{ for } d=1$$

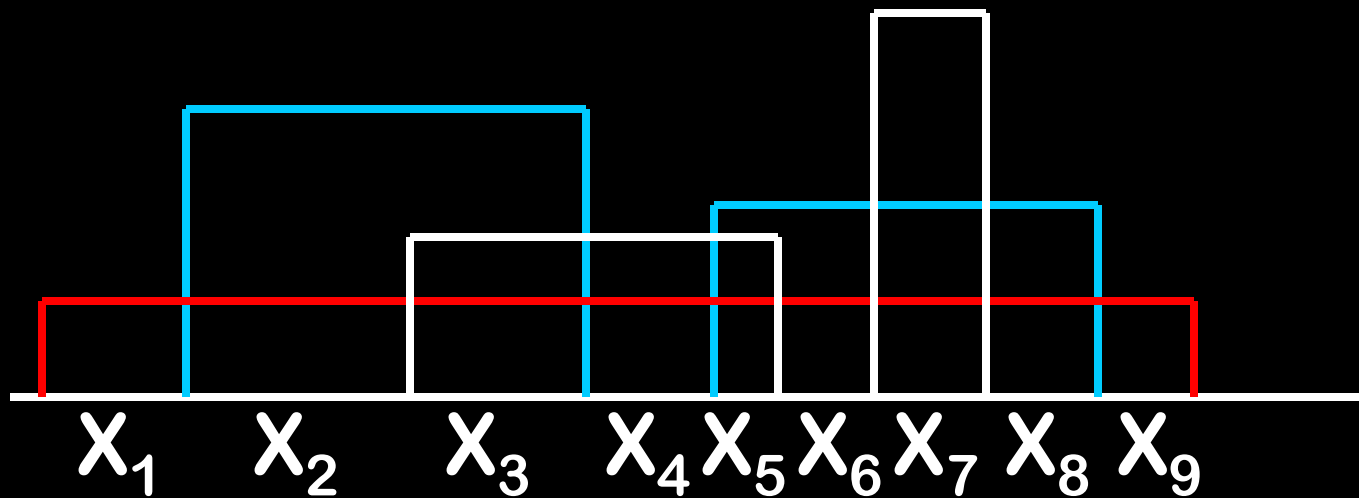
computing\* integrals **is hard**  $d > 1$

\* obtaining explicit expression for the density



What were we doing?

$$\int (f_1, f_2, f_3) dL = (w_1)_1, (w_2)_1, (w_3)_1$$



What were we doing?

$$\int (f_1, f_2, f_3) dL = (w_1)_1, (w_2)_1, (w_3)_1$$

Can we efficiently compute integrals  $dL$  for piecewise linear?

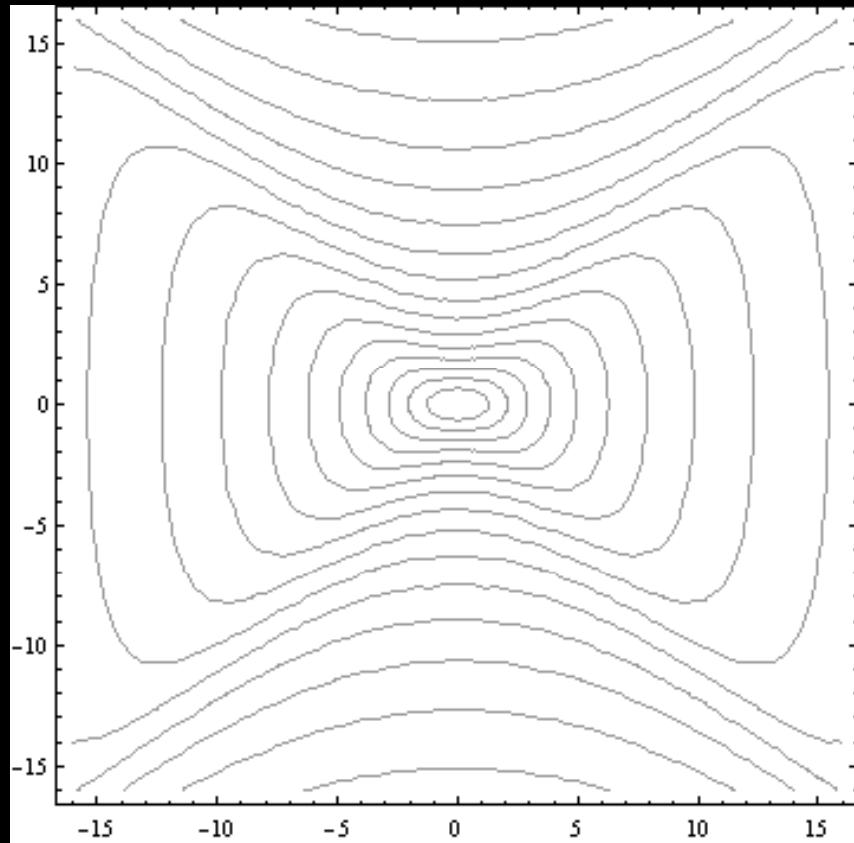


Can we efficiently compute  
integrals  $dL$  for piecewise linear?

$$\phi: \mathbb{R} \rightarrow \mathbb{R}^2$$

$$\phi(z) = (1, z)$$

$$(X, Y) = \int \phi \, dL$$



$$\phi: \mathbb{R} \rightarrow \mathbb{R}^2$$

$$\phi(z) = (1, z)$$

$$(X, Y) = \int \phi \, dL$$

$(2(X-Y), 2Y)$  has density at  $\frac{u+v, u-v}{2}$

$$\frac{4}{\pi((4+u^2)^2 + 16v^2)} + \Re \left( \frac{\pi + 2i \operatorname{arctanh}(v/\sqrt{4+u^2-4iv})}{2\pi(4+u^2-4iv)^{3/2}} \right)$$

All pairs  $L_1$ -distances for mixtures of uniform densities in time

$$O\left(\frac{(N^2 + Nn) (\log N)}{\varepsilon^2}\right)$$

All pairs  $L_1$ -distances for piecewise linear densities in time

$$O\left(\frac{(N^2 + Nn) (\log N)}{\varepsilon^2}\right)$$

# QUESTIONS

$$\phi: \mathbb{R} \rightarrow \mathbb{R}^3$$

1)  $\phi(z) = (1, z, z^2)$  ?

$$(X, Y, Z) = \int \phi \, dL$$

$$\frac{4}{\pi((4 + u^2)^2 + 16v^2)} + \Re \left( \frac{\pi + 2i \operatorname{arctanh}(v/\sqrt{4 + u^2 - 4iv})}{2\pi(4 + u^2 - 4iv)^{3/2}} \right)$$

2) higher dimensions ?

