Hierarchical Correlation Reconstruction — multi-directional propagation, of values/distrbutions, addtional training ways

HCRNN biology-inspired neural networks with neurons containing local joint distribution model as polynomial

Jarosław Duda, JU
arXiv:2405.05097
slides from th.if.uj.edu.pl/~dudaj/

2WNN

logical two-way(2W): ←⎍ or ⎍→ axon

of neighbors

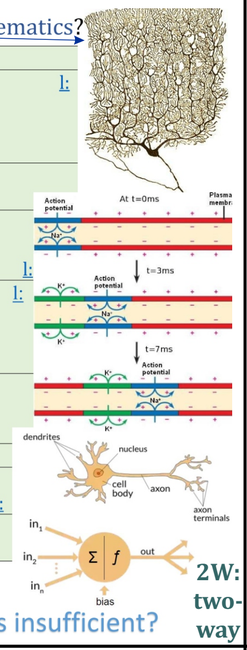
of normalized variables

represented by (a_j) parameters

1) **Biological neural networks (BNNs)** seem qualitatively superior to **current artificial (ANNs)**. **Biological neurons** have **thousands of connections**, also to itself, allowing to **hide complex mathematics**, also through **bursting** and **cancellation** of colliding action potentials. Instead of such complex **spiking NN physical level**, let us search for **higher level** "**logical neuron**" here: **extracting "biologically plausible" mathematics representing neuron dynamics.**

Spiking: **physical level**, but what is **logical**: hidden mathematics?

Parameters	ANN source	BNN
Structure	input weight output hidden layer	dendrites l: synapse axon cell body (~86B)
Learning	**very precise structures and formatted data**	**they can tolerate ambiguity**
Processor	complex high speed one or a few	simple low speed large number l:
Memory	**separate from a processor** localized non-content addressable	**integrated into processor** l: distributed content-addressable
Computing	centralized sequential **stored programs**	distributed paralel **self-learning**
Reliability	**very vulnerable**	**robust**
Expertise	numerical and symbolic manipulations	perceptual problems l:
Operating Environment	**well-defined well-constrained**	**poorly defined un-constrained**



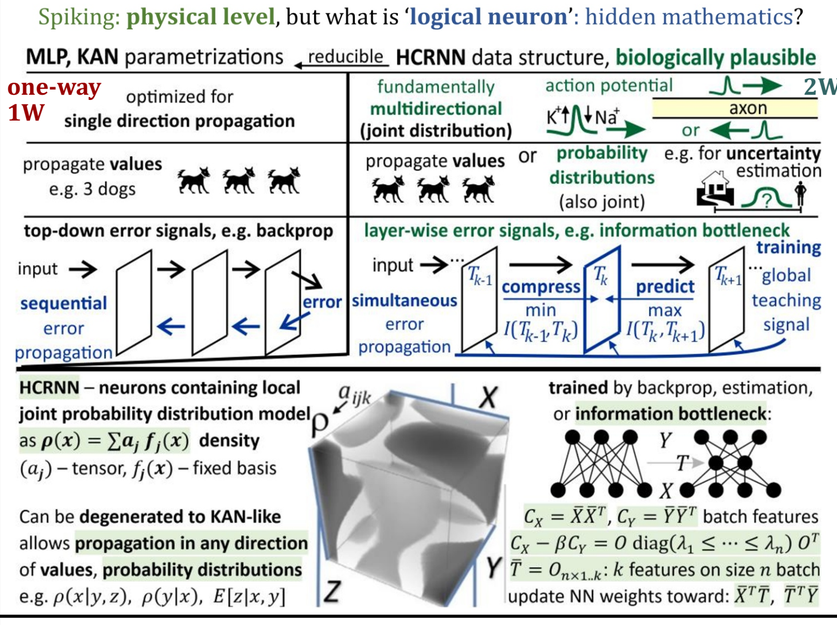
BNN learning ≠ ANN backpropagation

Qualitative superiority: just "take more neurons insufficient?

2W: two-way

2) While standard **logical neurons** like **MLP** (Multi-Layer Perceptron), **KAN** are **one-way (1W)**: focused on **uni-directional propagation**, **biological axons propagate in both directions** (**2W**) e.g. for training, forming **two-way neural networks (2WNN)**. Also value prediction alone seems insufficient for animals - it is crucial to estimate **uncertainty**, e.g. by **propagating entire distributions**. Additionally, **backpropagation** is viewed as **not biologically plausible** - biology needs **local layer-wise** training ways like **information bottleneck**.

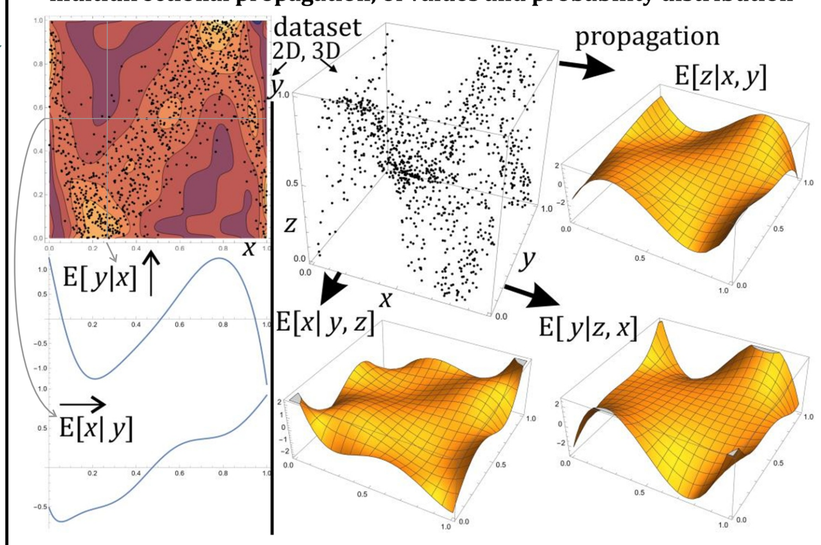
Spiking: **physical level**, but what is '**logical neuron**': hidden mathematics?



3) To catch up with **biology**, **2W logical neuron** can be built as **data structure** containing **model of local joint distribution**, e.g. among its connections. This way substituting some variables and normalizing would give **conditional distribution**, in **any direction**. Like in **Bayes** theorem but without it: straight from joint distribtuion. By averaging we could also use **inputs as distributions**, taking expected value (or median, mode) of output we can **propagate values**.

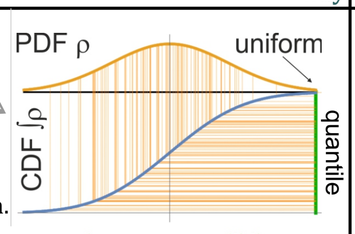
Joint distribution neurons as logical for BNNs:

multidirectional propagation, of values and probability distribution



just substitute and normalize to get conditional from joint distribution

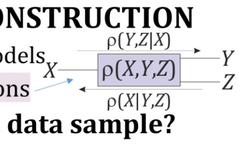
4) For **joint distribution data structure**, it is very convenient to **normalize variables to nearly uniform in [0,1]** (e.g. by $x\to \mathrm{CDF}(x)$ or EDF by just sorting), then **represent their joint density as just linear combination e.g. of polynomials**, preferably in **orthonormals basis** allowing for independent MSE estimation.



decompose dependencies into mixed moments – control, accuracy, credibilty

HIERARCHICAL CORRELATION RECONSTRUCTION

for time series, conditional distribution (~~Bayes~~) models, (nonlinear, adaptive, all-directional) artificial neurons



How to model/estimate density from a data sample?

MSE fit polynomial $\rho(x)=\sum_{f\in B} a_f f(x)$ (in (f) orthonormal basis)

also for **joint distribution**, **non-stationarity**, **missing data**

	Moments/cumulants	$\rho(x)=\sum_f a_f f(x)$	Machine learning, NN
# parameters	low – rough	from low to high	high - accurate
estimation	e.g. $m_k=\frac{1}{\lvert X\rvert}\sum_{x\in X} x^k$	$a_f=\frac{1}{\lvert X\rvert}\sum_{x\in X} f(x)$	usually iteration
Interpretable?	yes	Yes: mixed moments	depends, **trust**???
Independently?	yes	Yes (adapt, missing)	depends
Unique?	yes	yes (MSE)	often huge freedom
Accuracy?	controllable	controllable	usually uncontrollable
Density?	moment problem	**YES:** $\sum_f a_f f(x)$	depends
→ complete	depends	yes	depends

Jarek Duda

5) While **local basis like Gaussians** is useful for clustered data, for **neurons** there is sought e.g. XOR operation - trivial to realize with polynomial. Generally, **global basis** like **polynomials** usually gives **much better cross-validation like** below - **generalizes better**: searching for **general data description like moments** here, instead of just assuming that "new points will be close to old points" in local e.g. KDE or mixture models, also less convenient to represent.

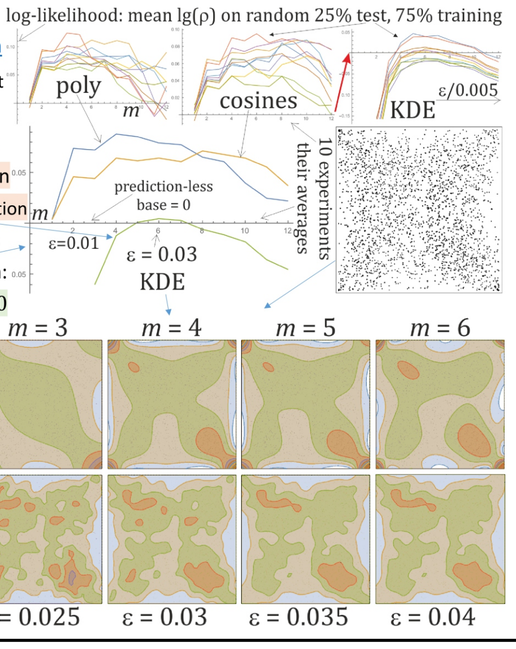
moments generalize well

KDE – kernel density estimation

e.g. ϵ-radius Gaussian in each point

- huge #parameters ~ #points
- how to choose (ellipse?) radii??
- doesn't work in high dimension
- terrible log-likelihood, generalization as it localizes in the old points

cross-validation:



Polynomial: MSE fitted to $\epsilon\to 0$

6) Basic formulas for **2D case** assuming **product basis** of **orthonormal** $(f^i(x))$ e.g. polynomials - with **(a_{ij}) as parameters contained by neuron** with 2 connections, generally as many indexes as connections. Substituting and normalizing we get formulas for propagation as conditional distributions, or their expected values, requiring to just **transpose** (permutate indexes) to **change propagation direction** - seems simple enough to fit in **biological**. Working on distributions we can also control **mutual information $I(X;Y)$**.

Neuron containing local joint distribution model – of its connections: (a_j) e.g. a_{ij} **matrix** $(d=2)$ of mixed moments **as neuron parameters**

propagation as conditional densities/expected values $E[x|y], E[y|x]$

$$\rho(x,y)=\sum_{i,j\ge 0} a_{ij} f_i(x) f_j(y)$$

(a_{ij}) in neuron $d=2$ model

$$a_{ij}=\frac{1}{|\bar{X}|}\sum_{(x,y)\in\bar{X}} f_i(x)\,f_j(y)$$

direct estimation

$$\rho(x|y)=\sum_{i\ge 0} f_i(x)\frac{\sum_j a_{ij}\ f_j(y)}{\sum_j a_{0j}\ f_j(y)}$$

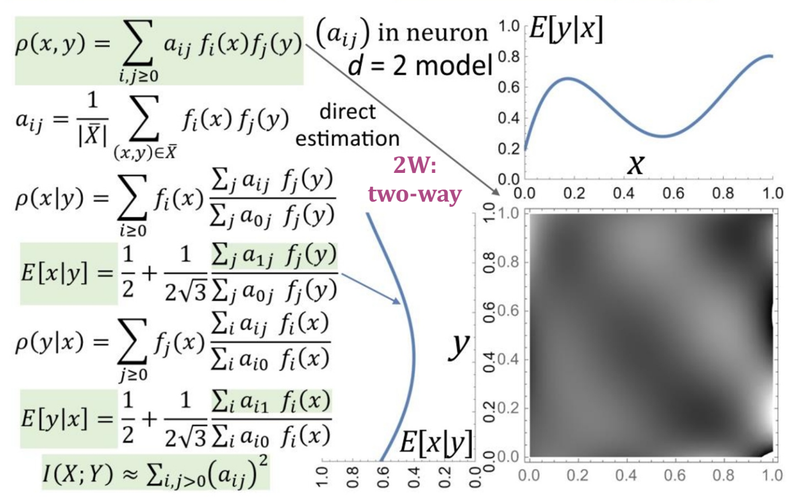
$$E[x|y]=\frac{1}{2}+\frac{1}{2\sqrt{3}}\frac{\sum_j a_{1j}\ f_j(y)}{\sum_j a_{0j}\ f_j(y)}$$

$$\rho(y|x)=\sum_{j\ge 0} f_j(x)\frac{\sum_i a_{ij}\ f_i(x)}{\sum_i a_{i0}\ f_i(x)}$$

$$E[y|x]=\frac{1}{2}+\frac{1}{2\sqrt{3}}\frac{\sum_i a_{i1}\ f_i(x)}{\sum_i a_{i0}\ f_i(x)}$$

$$I(X;Y)\approx\sum_{i,j>0}(a_{ij})^2$$

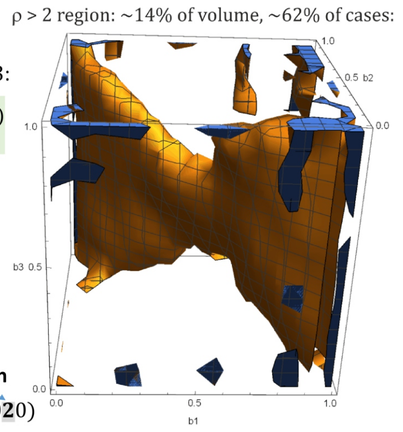
2W: two-way



7) First [0,1] (Legendre) **orthonormal polynomials** and their **product basis** for 2,3 variables, allowing for **hierarchical reconstruction of dependencies**. As $f_0=1$, **coordinates being 0 control normalization**, there is **no dependence** with such variables. Coefficients with single nonzero index describe **marginal distributions** of variables, with two nonzero define **pairwise dependencies**, with 3 **triplewise**, and so on **hierarchically** adding higher order dependencies.

8) Analogously in ***d* dimensions** we need ***d* index tensor representing joint distribution**. Given a_j coefficient describes dependencies between variables of nonzero indexes of $\boldsymbol{j}$ - in practice requiring to **restrict** e.g. to only pairwise dependencies up to e.g. 4th moment (kurtosis). MSE **estimation** by **averaging** can use **exponential moving average** instead for **adaptivity**. The 0 index values can be **missing**, allowing to **extract all available dependencies**.

ρ > 2 region: ~14% of volume, ~62% of cases:



Also $[0,1]^d$ in **higher dimensions**, e.g. $d=3$:

$$\rho(x_1,x_2,x_3)=\sum_{j\in B} a_j\ f_{j_1}(x_1)f_{j_2}(x_2)f_{j_3}(x_3)$$

⇒ **conditional distributions without Bayes**

MSE estimated from dataset $X\subset\mathbb{R}^3$:

$$a_j=\frac{1}{|X|}\sum_{x\in X} f_{j_1}(x_1)f_{j_2}(x_2)f_{j_3}(x_3)$$

For considered **statistical dependencies**:

basis B of **considered mixed moments**

a_j describes e.g. **variance-variance between hierarchical** e.g. $d=9$: $B\ni \boldsymbol{j}=(000200020)$

a_j = **average** of $f_{j_1}(x_1)\dots f_{j_d}(x_d)$ over dataset - entire or:

- over a subset for **missing data** - we need only $j>0$ coordinates as $f_0=1$
- $a_j^{t+1}=\lambda a_j+(1-\lambda)f_j(\boldsymbol{x})$ parameter evolution for **nonstationary time series**

further statistical dependencies

pair-wise joint density ≈ independent + a_{11} · ~ correlation coef. + a_{12} · + a_{21} · + a_{22} · var-var

9) Can **degenerate** to very popular **KAN: Kolmogorov-Arnold networks** if **only pairwise dependecies**, offering **many improvements**.t

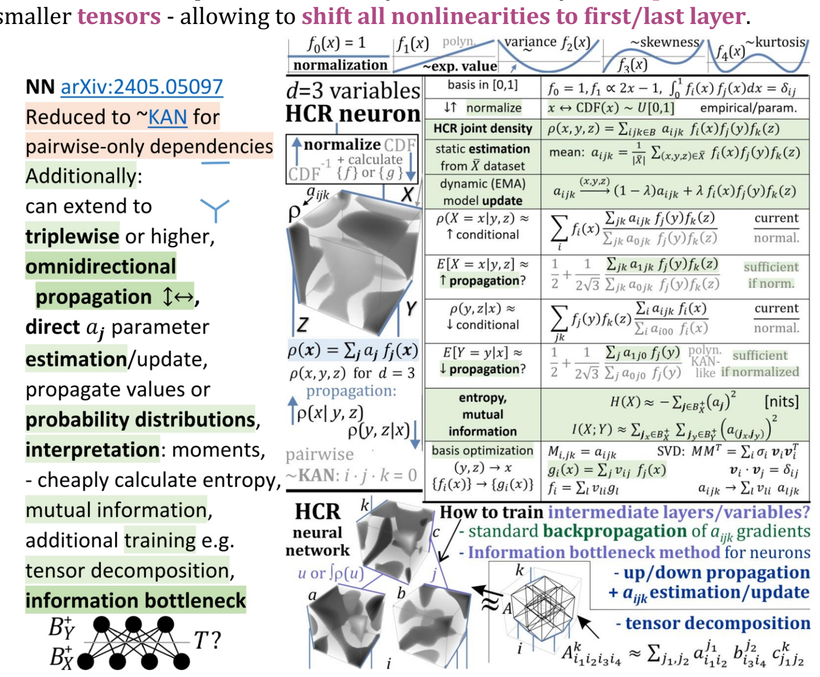
HCRNN: ~**KAN**-like parametrization, but with **many advantages**

- can **propagate in any direction** like **biological neural networks**,
- **propagate** values or **probability distributions**,
- **interpretation** of parameters as **mixed moments,**
- consciously add **triplewise** and higher dependencies,
- inexpensive evaluation of modeled **mutual information**,
- additional **training** approaches, e.g. **direct estimation**, **tensor decomposition**, **information bottleneck** of hidden layer

```
f[x_] := Exp[x[[1]] - x[[2]]^2 - x[[3]]^3 + x[[4]]^4]; d = 4; (* function, variables *)
m = 4; n = 1000; SeedRandom[1]; (* max. features, points, next find basis: *)
fs = Expand[Table[LegendreP[i, x] * Sqrt[(2 i + 1)] /. x -> 2 x - 1, {i, m}]];
X = RandomVariate[NormalDistribution[0, 1], {n, d}]; Y = Map[f, X];
inX = Table[normEDF[X[[All, i]], {i, d}]; inY = normEDF[Y]; (*normalization*)
nX = Transpose[Table[inX[[i]][X[[All, i]], {i, d}]]; nY = inY[Y];
md = 4; X̄ = Flatten[fs[[1 ;; md]] /. x -> nX, {{2}, {3, 1}}]; (* features *)
mu = 1; Ȳ = Transpose[fs[[1 ;; mu]] /. x -> nY]; (* only expected value *)
as = Flatten[Transpose[X̄].Ȳ] / n; (* direct parameter estimation *)
cp = 1 / 2 + Expand[Partition[as, md].fs[[1 ;; md]] / Sqrt[12]; (* E *)
Print[Row[{ListPlot[Transpose[{nY, Y}]], "of sum of below: "}]];
Row[Table[cf = cp[[i]] /. x -> inX[[i]][x]; (* polynomials(normalized X) *)
  Plot[cf, {x, -2, 2}, ImageSize -> 130], {i, d}]]
```

4+1 [0,1] joint distr. model

trained



10) **Neurons containing joint distribution** also allow for online control of often the most valuable e.g. for **training** and **explainability**: (conditional) **entropy, mutual information evaluation** - in just **bits of information** one variable says about another, **invariant** to permutations and bijections. For HCR it can be **approximated** with just **sums of squares** of our moments/neuron parameters: nontrivial between variables of interest. Together with MSE estimation of (a) coefficients from dataset, we get simple **HSIC**-like formula.

Information theory [nats ≡ bits/ln(2)]:

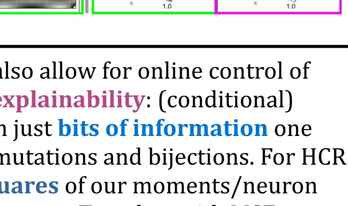
entropy, mutual information for HCR

$$\rho(x)=\sum_{j\in B} a_j f_j(x)=1+\sum_{j\in B^+} a_j f_j(x)$$

base — without **0**

Entropy with $\ln(1+\rho)\approx\rho+O(\rho^2)$ approx.:

$$H(X)=-\int_{[0,1]^d}\rho(x)\ln(\rho(x))\,dx\approx-\int\left(\sum_{j\in B} a_j f_j(x)\right)\left(\sum_{j\in B^+} a_j f_j(x)\right)dx=-\sum_{j\in B^+}(a_j)^2$$



joint: $H(X,Y)\approx-\sum_{(j_x,j_y)\in B_X\times B_Y\setminus\{0\}}\left(a_{(j_x,j_y)}\right)^2$

conditional:

$$H(X|Y)=H(X,Y)-H(Y)\approx-\sum_{j_X\in B_X^+}\sum_{j_Y\in B_Y}\left(a_{(j_x,j_y)}\right)^2$$

Mutual information: how many **bits** X says about Y

evaluation invariant to permutations, bijections

$$I(X;Y)=H(X)+H(Y)-H(X,Y)\approx\sum_{j_X\in B_X^+}\sum_{j_Y\in B_Y^+}\left(a_{(j_x,j_y)}\right)^2$$

nontrivial X-Y moments:

With estimation from features: $\bar{X}_{i=1..n,\,j\in B_X^+}=\frac{f_j(x^i)}{\sqrt{n}}$

MSE:

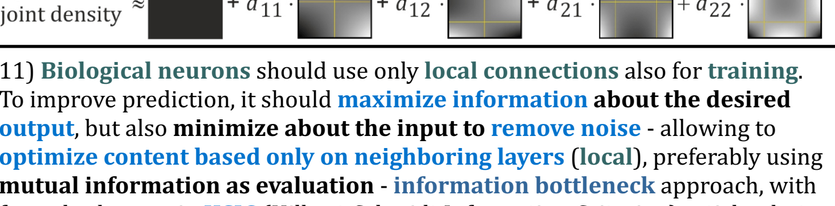
$\boldsymbol{a}=\bar{X}^T\bar{Y}$ **getting ~**HSIC: $I(X;Y)\approx\mathrm{Tr}\left((\bar{X}\bar{X}^T)(\bar{Y}\bar{Y}^T)\right)$

11) **Biological neurons** should use only **local connections** also for **training**. To improve prediction, it should **maximize information about the desired output**, but also **minimize about the input to remove noise** - allowing to **optimize content based only on neighboring layers (local)**, preferably using **mutual information as evaluation** - **information bottleneck** approach, with formulas here as in **HSIC** (Hilbert-Schmidt Information Criterion) articles, but replacing their **local basis** of Gaussians, with **global** of polynomials/moments.

Naftali Tishby, **information theoretic view** – permutation, bijection independent

Markov process between layers, **first extract/compress essential information** reducing **mutual information** [**bits**] $H(X)\ge I(X;T_1)\ge I(X;T_2)\ge\cdots$

Information bottleneck (Tishby): for $X\to T\to Y$ optimize $\inf_T I(X;T)-\beta I(T;Y)$



$I\sim\mathrm{HSIC}=\mathrm{Tr}(K_x,K_y)$ $\quad K_x(x,x')=\langle\phi(x),\phi(x')\rangle$ $\quad n\times n$ as **sample size**

ϕ Gaussian: **local, width?** HCR: **many global** poly (f_j) **for normalized**

12) **Summary** with formulas for 3 variables. While single neuron can be **directly trained** or **updated**, for **intermediate layers it is more difficult**, but we can always treat such network as just **KAN-like parametrization** and **train with gradient descent**. One of additional training options is estimating **high order tensor** as dependencies, and try to automatically **decompose** it into smaller **tensors** - allowing to **shift all nonlinearities to first/last layer**.

NN arXiv:2405.05097

Reduced to ~KAN for pairwise-only dependencies

Additionally:

can extend to **triplewise** or higher, **omnidirectional propagation ↕↔**, **direct** a_j parameter **estimation**/update, propagate values or **probability distributions**, **interpretation**: moments, - cheaply calculate entropy, mutual information, additional training e.g. tensor decomposition, **information bottleneck**

basis in [0,1]	$f_0=1, f_1\propto 2x-1,\ \int_0^1 f_i(x)\,f_j(x)dx=\delta_{ij}$
↓↑ normalize	$x\leftrightarrow \mathrm{CDF}(x)\sim U[0,1]$ empirical/param.
HCR joint density	$\rho(x,y,z)=\sum_{ijk\in B} a_{ijk}\ f_i(x)f_j(y)f_k(z)$
static **estimation** from $\bar{X}$ dataset	mean: $a_{ijk}=\frac{1}{\lvert\bar{X}\rvert}\sum_{(x,y,z)\in\bar{X}} f_i(x)f_j(y)f_k(z)$
dynamic (EMA) model **update**	$a_{ijk}\xrightarrow{(x,y,z)}(1-\lambda)a_{ijk}+\lambda\,f_i(x)f_j(y)f_k(z)$
$\rho(X=x\|y,z)\approx$ ↑ conditional	$\sum_i f_i(x)\frac{\sum_{jk} a_{ijk}\ f_j(y)f_k(z)}{\sum_{jk} a_{0jk}\ f_j(y)f_k(z)}$ current / normal.
$E[X=x\|y,z]\approx$ ↑ **propagation?**	$\frac{1}{2}+\frac{1}{2\sqrt{3}}\frac{\sum_{jk} a_{1jk}\ f_j(y)f_k(z)}{\sum_{jk} a_{0jk}\ f_j(y)f_k(z)}$ sufficient if norm.
$\rho(y,z\|x)\approx$ ↓ conditional	$\sum_{jk} f_j(y)f_k(z)\frac{\sum_i a_{ijk}\ f_i(x)}{\sum_i a_{i00}\ f_i(x)}$ current / normal.
$E[Y=y\|x]\approx$ ↓ **propagation?**	$\frac{1}{2}+\frac{1}{2\sqrt{3}}\frac{\sum_j a_{1j0}\ f_j(y)}{\sum_j a_{0j0}\ f_j(y)}$ polyn. KAN-like sufficient if normalized
entropy, mutual information	$H(X)\approx-\sum_{I\in B_X^+}(a_I)^2$ [nits] $I(X;Y)\approx\sum_{I_x\in B_X^+}\sum_{I_Y\in B_Y^+}\left(a_{(I_x,I_y)}\right)^2$
basis optimization $(y,z)\to x$ $\{f_i(x)\}\to\{g_i(x)\}$	$M_{i,jk}=a_{ijk}$ SVD: $MM^T=\sum_i\sigma_i\boldsymbol{v}_i\boldsymbol{v}_i^T$ $g_i(x)=\sum_j v_{ij}\ f_j(x)$ $\quad \boldsymbol{v}_i\cdot\boldsymbol{v}_j=\delta_{ij}$ $f_i=\sum_l v_{il}\,g_l$ $\quad a_{ijk}\to\sum_l v_{il}\ a_{ijk}$

B_Y^+ B_X^+ T?