Organizing higher-order cliques by sparse representation

Hiroshi Ishikawa

Nagoya City University

GC2008: Graph Cuts and Related Discrete or Continuous Optimization Problems. February 25 - 29, 2008 at IPAM, UCLA

Example: Binary Restoration

original

noise added Y_v

restored X_v

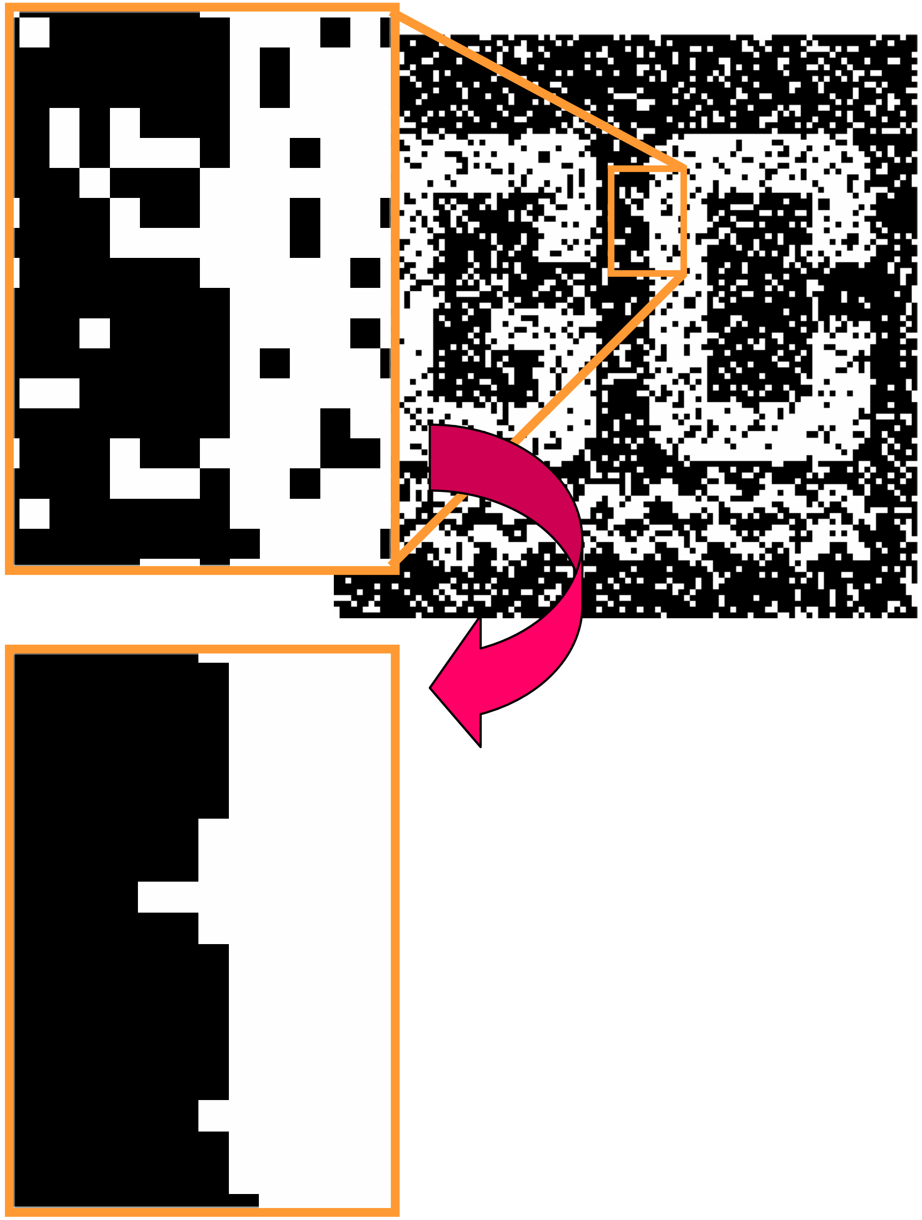
Greig, Porteous and Seheult '89

$$E(X) = \sum_{v \in V} \lambda | Y_v - X_v | + \sum_{(u,v) \in E} | X_u - X_v |$$

- This is the optimal solution, given the smoothing prior.
- But it does not look as perfect as it sounds.
- We can probably do a better job removing the noise by hand. How?

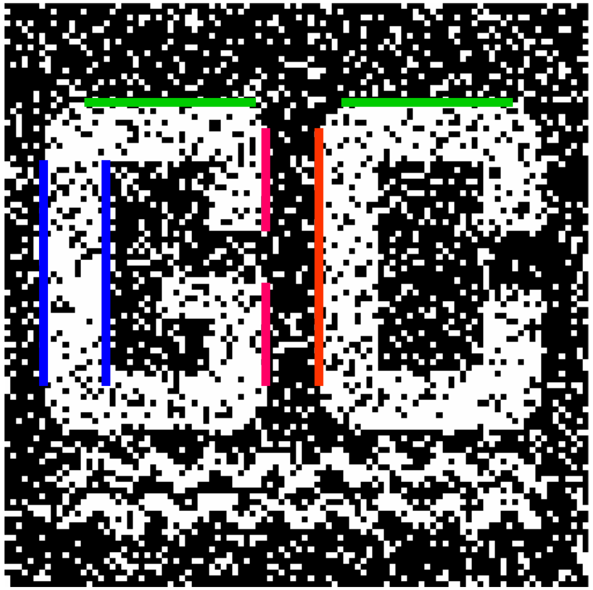
Higher-order Relationships

- Boundary smoothness



Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism



Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism
- Larger structure

Higher-order Relationships

- Boundary smoothness
- Linearity
- Collinearity
- Parallelism
- Larger structure
- Regular structure

Higher-order Relationships

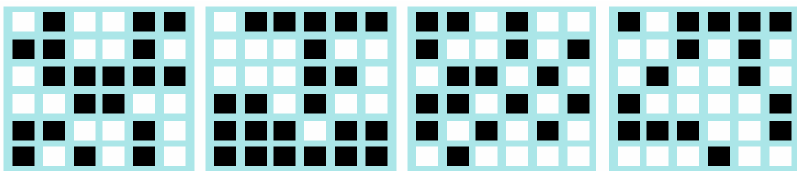
- Boundary smoothness
- Linearity
- Collinearity
- Parallelism
- Larger structure
- Regular structure
- Higher-level knowledge

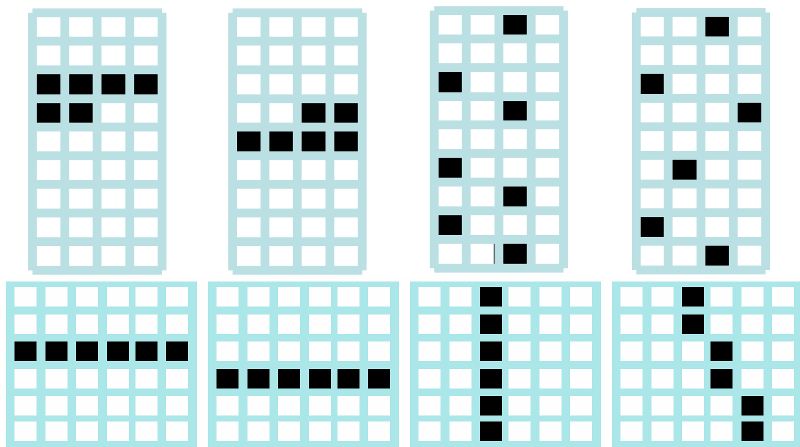
- These tell us what is wrong with ⇒

Can we state the problem?

- How can we state the problem precisely?
- What space do these things live in?
  - First order: V
  - Second order: $V \times V$

  …… *exponential*
- Simple ones matter more
- Simplicity is relative to the structure





Lesson from the graph

- By talking about graphs, we are aware explicitly what structure we are using
- That makes the structure explicit and manipulable
- Graph abstracts only the neighborhood structure
- How can we abstract other structures?

Approach

- Represent patterns in a uniform way
- Reflect simplicity relative to the structure
- Integrate the low level and the high level
- Automatically generate relationships to look for
- Learn probabilities from data —Prior
- Connection to the signal level —Semantics
  - To generate structures automatically, this semantics must be a part of what is generated.

Dense representation

- Something like bitmap
  - Can represent anything but does not reflect any structure
  - Most that can be represented this way is random noise
- Abstract a bit → subsets
  - Lines, circles, and other geometric objects
  - Images can be thought of as a subset of $\mathbf{R}^2 \times Color$

    i.e., image function $I(p)$ on $D \subset \mathbf{R}^2 \Leftrightarrow \{(p, I(p)) \mid p \in D\}$
- We call this the dense representation
- The representation in the following includes these as the trivial representation
- It also allows sparser representation of the same subset if it is simple relative to the structure of the space

Representation

- Specify sets and *power maps* between them (*diagram*)

Ex.

$$\varphi : \mathcal{P}(Y) \to \mathcal{P}(X)$$
$$\psi : \mathcal{P}(Z) \to \mathcal{P}(Y)$$
$$\eta : \mathcal{P}(W) \to \mathcal{P}(Y)$$
$$\kappa : \mathcal{P}(X) \to \mathcal{P}(W)$$

$$\begin{tikzcd}
 & X \arrow[drr, dotted, "\kappa"'] & & Y \arrow[ll, dotted, "\varphi"'] & & Z \arrow[ll, dotted, "\psi"'] & \\
s(X) \arrow[ur, hook] & & & W \arrow[u, dotted, "\eta"'] & s(Y) \arrow[ul, hook] & & s(Z) \arrow[ul, hook] \\
 & & & & s(W) \arrow[ul, hook] & &
\end{tikzcd}$$

- Consider an assignment s to each set of its subset that satisfies

$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

$$s(X) = \varphi(s(Y))$$
$$s(Y) = \psi(s(Z)) \cap \eta(s(W))$$
$$s(W) = \varphi(s(X))$$

where $\mathrm{in}(S)$: incoming power maps to S

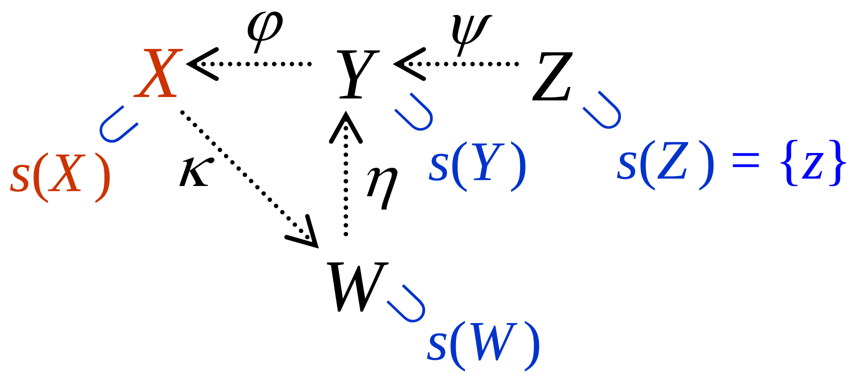
$$\mu : \mathcal{P}(\mathrm{dm}(\mu)) \to \mathcal{P}(\mathrm{cdm}(\mu))$$

Representation

- Specify sets and *power maps* between them (*diagram*)

Ex.

$$\varphi : \mathcal{P}(Y) \to \mathcal{P}(X)$$
$$\psi : \mathcal{P}(Z) \to \mathcal{P}(Y)$$
$$\eta : \mathcal{P}(W) \to \mathcal{P}(Y)$$
$$\kappa : \mathcal{P}(X) \to \mathcal{P}(W)$$



- Consider an assignment s to each set of its subset

  that satisfies

$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

$$s(X) = \varphi(s(Y))$$
$$s(Y) = \psi(s(Z)) \cap \eta(s(W))$$
$$s(W) = \varphi(s(X))$$

- Finally, specify some of the s(.)
- Designate one set (say X) whose assigned subset ($s(X)$) is the dense representation of what is represented.

The Simplest Examples

$$\begin{tikzcd} s(\mathbf{R})=\{r\} \arrow[r, phantom, "\subset"] & \mathbf{R} \arrow[rr, dotted, "\mathrm{dist}^{-1}"] & & X\times X' \arrow[d, dotted, "\pi_1"] \\ s(X')=\{p\} \arrow[r, phantom, "\subset"] & X' \arrow[urr, dotted, "\pi_2^{-1}"] & s(X)=? \arrow[r, phantom, "\subset"] & X \end{tikzcd}$$

X, X' : Euclidean plane

dist: $X\times X\to\mathbf{R}$ distance function

π_1, π_2: Projection

$$s(S)=\bigcap_{\mu\in\mathrm{in}(S)}\mu(s(\mathrm{dm}(\mu)))$$

$f: A\to B$

$\longrightarrow$

$f: \mathcal{P}(A)\to\mathcal{P}(B) \qquad A\supset S\mapsto\{f(x)\mid x\in S\}\subset B$

$f^{-1}: \mathcal{P}(B)\to\mathcal{P}(A) \qquad B\supset S\mapsto\{x\in A\mid f(x)\in S\}\subset A$

The Simplest Examples

$$
\begin{tikzcd}
s(\mathbf{R})=\{r\} \arrow[r, phantom, "\subset"] & \mathbf{R} \arrow[rr, dotted, "\mathrm{dist}^{-1}"] & & X\times X' \arrow[d, dotted, "\pi_1"] \\
s(X')=\{p\} \arrow[r, phantom, "\subset"] & X' \arrow[urr, dotted, "\pi_2^{-1}"] & & X \\
& & s(X)=? \arrow[ur, phantom, "\subset"] &
\end{tikzcd}
$$

X, X' : Euclidean plane

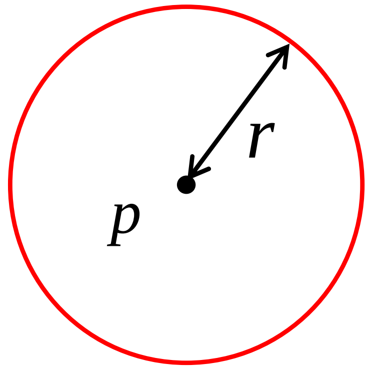
dist: $X\times X\to\mathbf{R}$ distance function

π_1, π_2: Projection

$$s(S)=\bigcap_{\mu\in \mathrm{in}(S)}\mu(s(\mathrm{dm}(\mu)))$$

$$
\begin{aligned}
s(X\times X') &= \mathrm{dist}^{-1}(s(\mathbf{R}))\cap\pi_2^{-1}(s(X')) \\
&= \{(x,y)\mid \mathrm{dist}(x,y)\in s(\mathbf{R}), \pi_2(x,y)=p\} \\
&= \{(x,p)\mid \mathrm{dist}(x,p)=r\}
\end{aligned}
$$

$$
\begin{aligned}
s(X) &= \pi_1(s(X\times X')) \\
&= \{x\mid \mathrm{dist}(x,p)=r\}
\end{aligned}
$$



The Simplest Examples

$$\begin{tikzcd} V \arrow[r, dotted, "\pi_1^{-1}"] & V\times\mathbf{R} \arrow[r, dotted, "\mathrm{mult}"] & V' \arrow[dl, dotted, "\mathrm{sub}^{-1}"'] \\ X' \arrow[r, dotted, "\pi_2^{-1}"] & X\times X' \arrow[r, dotted, "\pi_1"] & X \end{tikzcd}$$

$s(V)=\{v\} \subset V$

$s(X')=\{p\} \subset X'$

$s(X)=?$, $s(X) \subset X$

X, X' : Euclidean plane

V, V' : 2D vector space

mult: $V\times\mathbf{R}\to V \qquad (v,c)\mapsto cv$

sub: $X\times X\to V \qquad (x,y)\mapsto x-y$

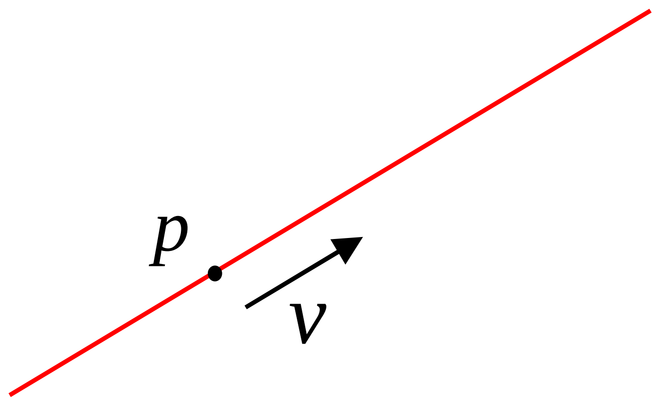
π_1, π_2: Projection

$$s(V\times\mathbf{R}) = \{(v,c)\mid c\in\mathbf{R}\}$$

$$s(V') = \{cv\mid c\in\mathbf{R}\}$$

$$s(X\times X') = \{(x,p)\mid x-p\in s(V')\}$$

$$s(X) = \{x\mid x=p+cv, c\in\mathbf{R}\}$$



Remarks

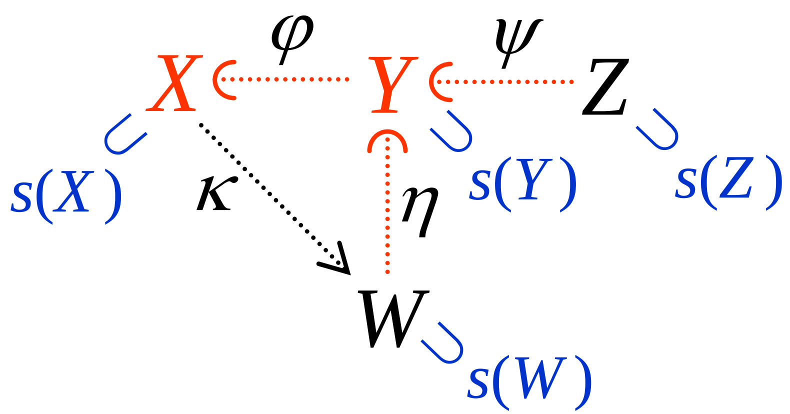
- The diagram (the sets and the maps) represents the structure; the fixed subsets are the parameters
- Not too many primitive sets and maps
- Variety comes from combination
- Once the primitive sets and maps are implemented, the rest is automatic
- The structure and implementation details can be separated; e.g., real numbers, image resolution, …
  - A line is a line in this representation, no matter how the points on the Euclidean plane is represented.

A little extension

- Choose some of the sets

Ex.

$$\varphi : \mathcal{P}(Y) \to \mathcal{P}(X)$$
$$\psi : \mathcal{P}(Z) \to \mathcal{P}(Y)$$
$$\eta : \mathcal{P}(W) \to \mathcal{P}(Y)$$
$$\kappa : \mathcal{P}(X) \to \mathcal{P}(W)$$



- Assume the assignment s satisfies for the chosen sets

$$s(S) = \bigcup_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

$$s(X) = \varphi(s(Y))$$
$$s(Y) = \psi(s(Z)) \cup \eta(s(W))$$
$$s(W) = \varphi(s(X))$$

instead of

$$s(S) = \bigcap_{\mu \in \mathrm{in}(S)} \mu(s(\mathrm{dm}(\mu)))$$

Formally,

Consider the triple $(\mathscr{S}, \mathscr{S}', \mathscr{M})$ where

$\mathscr{S} = \{S_i\}_{i \in I}$ is a family of sets indexed by a set I

$\mathscr{S}'$ is a subfamily of $\mathscr{S}$, (chosen sets)

$\mathscr{M} = \{\varphi_j\}_{j \in J}$ is a family of maps $\varphi_j : \mathcal{P}(S_i) \to \mathcal{P}(S_k)$

Let s be an assignment to each S_i in $\mathscr{S}$ of $s_i \subset S_i$

that satisfies $s_i = \bigcup_{j \in J, \varphi_j : \mathcal{P}(S_k) \to \mathcal{P}(S_i)} \varphi_j(s_k)$ if S_i is in $\mathscr{S}'$

and $s_i = \bigcap_{j \in J, \varphi_j : \mathcal{P}(S_k) \to \mathcal{P}(S_i)} \varphi_j(s_k)$ if S_i is in $\mathscr{S} - \mathscr{S}'$

Combination

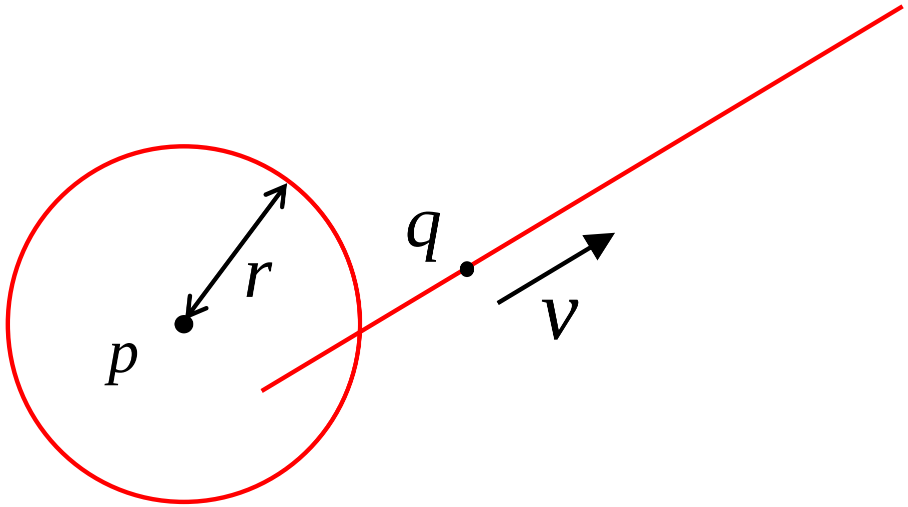
$$\begin{tikzcd}
\mathbf{R} \arrow[r, dotted, "\mathrm{dist}^{-1}"] & X\times X' \arrow[ddd, dotted, hook, "\pi_1"] & V \arrow[r, dotted, "\pi_1^{-1}"] & V\times\mathbf{R} \arrow[r, dotted, "\mathrm{mult}"] & V' \arrow[ddl, dotted, "\mathrm{sub}^{-1}"'] \\
X' \arrow[ur, dotted, "\pi_2^{-1}"] & & & & \\
 & & X' \arrow[r, dotted, "\pi_2^{-1}"] & X\times X' \arrow[dll, dotted, hook', "\pi_1"] & \\
 & X & & &
\end{tikzcd}$$

$s(\mathbf{R})=\{r\}$

$s(X')=\{p\}$

$s(V)=\{v\}$

$s(X')=\{q\}$



Recursive definition

$$\begin{tikzcd}
 & V \arrow[r, dotted, "\pi_2^{-1}"] & X\times V \arrow[d, dotted, bend left, "\mathrm{add}"] \\
s(V)=\{v\} \arrow[ur, phantom, "\subset"] & X' \arrow[r, dotted, "\mathrm{id.}"] & X \arrow[u, dotted, "\pi_1^{-1}"] \\
s(X')=\{p\} \arrow[ur, phantom, "\subset"] & & s(X)=? \arrow[u, phantom, "\subset"]
\end{tikzcd}$$

X, X' : Euclidean plane

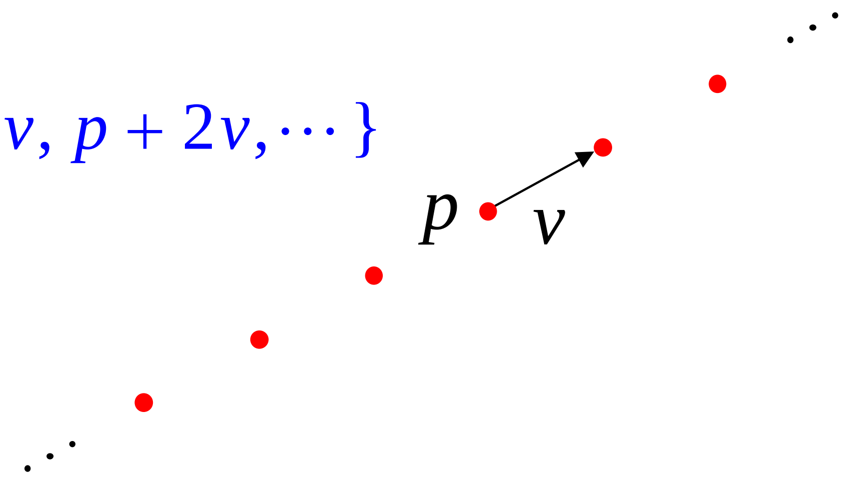
V: 2D vector space

add: $X\times V \longrightarrow X \quad (x,v)\mapsto x+v$

π_1, π_2: Projection

$$s(X\times V) = \{(x,v) \mid x\in s(X)\}$$

$$\begin{aligned} s(X) &= s(X')\cup \mathrm{add}(s(X\times V)) \\ &= \{p\}\cup\{x+v \mid x\in s(X)\} \\ &\supset \{\cdots, p-2v, p-v, p, p+v, p+2v, \cdots\} \end{aligned}$$



Recursive definition

$$
\begin{tikzcd}
V \arrow[r, dotted, "\pi_2^{-1}"] & X \times V \arrow[d, dotted, shift left, "\mathrm{add}"] \\
X' \arrow[r, dotted, "\mathrm{id.}"] & X \arrow[u, dotted, shift left, "\pi_1^{-1}"]
\end{tikzcd}
$$

$s(V) = \{u, v\} \subset V$

$s(X') = \{p\} \subset X'$

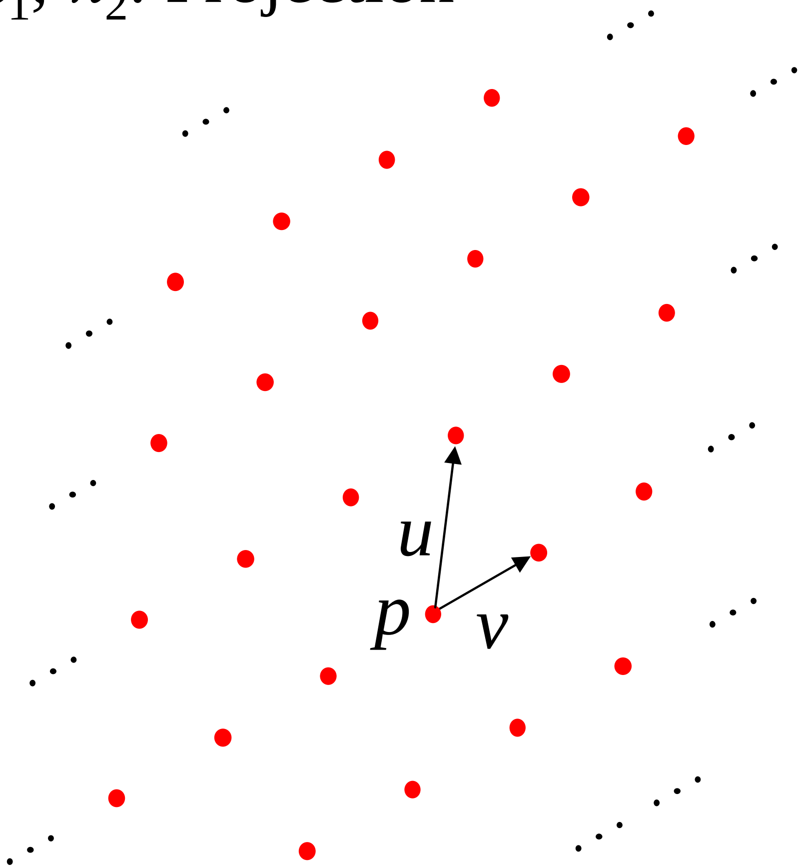
$s(X) = ?$, $s(X) \subset X$

X, X' : Euclidean plane

V: 2D vector space

add: $X \times V \longrightarrow X \quad (x, v) \mapsto x + v$

π_1, π_2: Projection



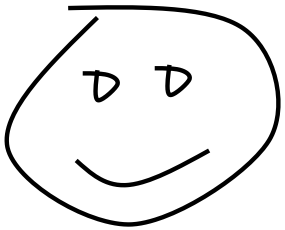
Hierarchical definition

$$\begin{tikzcd} V \arrow[r, dotted, "\pi_2^{-1}"] & X\times V \arrow[d, dotted, shift left, "\mathrm{add}"] \\ X' \arrow[r, dotted, "\mathrm{id.}"] & X \arrow[u, dotted, shift left, "\pi_1^{-1}"] \end{tikzcd}$$

$s(V) = \{u, v\}$

$s(X') = A$

$s(X) = ?$



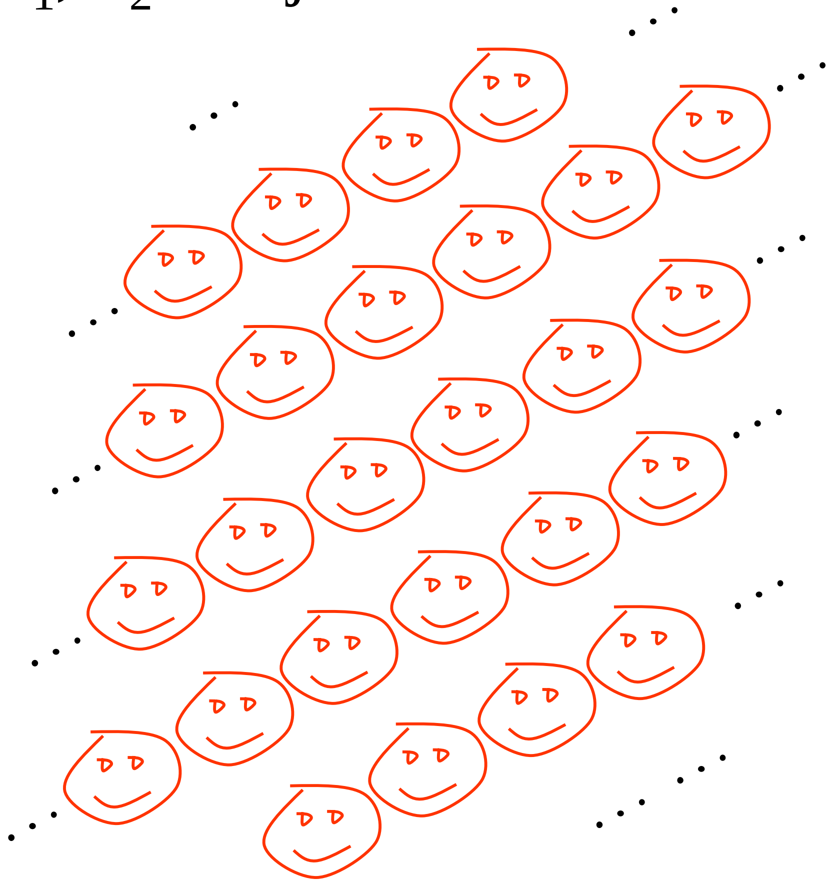
$A \subset X$

X, X' : Euclidean plane

V: 2D vector space

add: $X\times V \longrightarrow X \quad (x, v) \mapsto x + v$

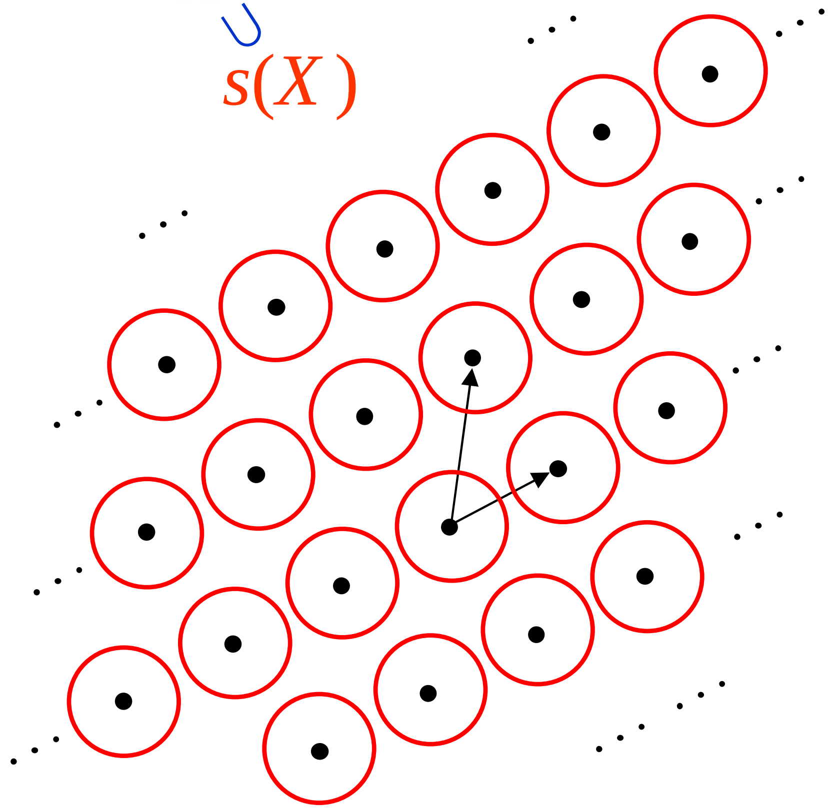
π_1, π_2: Projection



Hierarchical definition

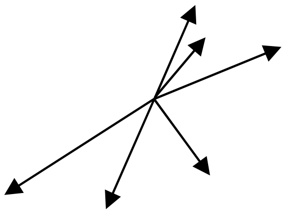
$$\begin{tikzcd}
s(\mathbf{R})=\{r\} \arrow[r, phantom, "\subset"] & \mathbf{R} \arrow[r, dotted, "\mathrm{dist}^{-1}"] & X\times X' \arrow[dr, dotted, "\pi_1"] & \\
s(X')=\{p\} \arrow[r, phantom, "\subset"] & X' \arrow[ur, dotted, "\pi_2^{-1}"] & & X'
\end{tikzcd}$$

$$\begin{tikzcd}
\{u,v\} \arrow[r, phantom, "\subset"] & V \arrow[r, dotted, "\pi_2^{-1}"] & X\times V \arrow[d, dotted, shift left, "\mathrm{add}"] \\
& X' \arrow[r, dotted, hook, "\mathrm{id.}"] & X \arrow[u, dotted, shift left, "\pi_1^{-1}"] \arrow[r, phantom, "\supset"] & s(X)
\end{tikzcd}$$

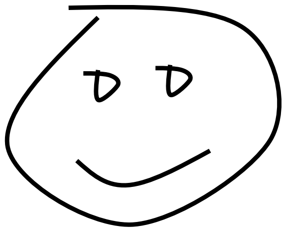


Regular and random parts of data

- Example: Same pattern appearing randomly
  - Regularity: the sameness of the pattern
  - Randomness: pattern itself & where the pattern appears

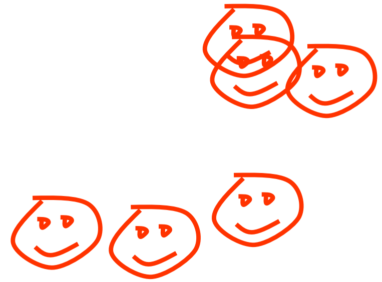


$R \subset V$



$A \subset X$

$$\begin{tikzcd}
 & V \arrow[r, dotted, "\pi_2^{-1}"] & X \times V \arrow[d, dotted, "\text{add}"] \\
R \arrow[ur, phantom, "\subset" sloped] & X' \arrow[ur, dotted, "\pi_1^{-1}"] & X \\
A \arrow[ur, phantom, "\subset" sloped] & & s(X) = ? \arrow[u, phantom, "\supset" sloped]
\end{tikzcd}$$



Representing Computation

- Can represent computation

$$\begin{tikzcd} \mathbf{N}\times\mathbf{N} \arrow[r, dotted] & \mathbf{N}\times\mathbf{N} \arrow[loop right, dotted, "(n,m)\mapsto(n+1,m(n+1))"] \end{tikzcd}$$

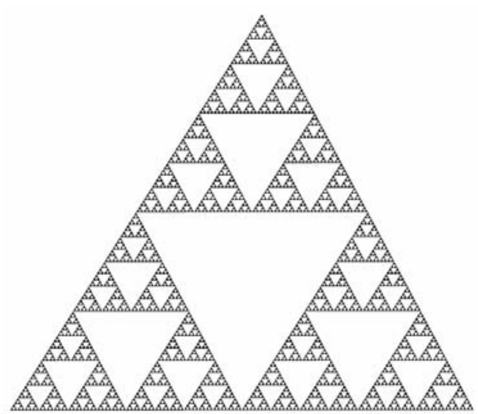
$$s(\mathbf{N}\times\mathbf{N})=\{(0,1)\} \subseteq \mathbf{N}\times\mathbf{N} \qquad \mathbf{N}\times\mathbf{N} \supseteq \{(0,1),(1,1),(2,2),(3,6),\cdots,(n,n!),\cdots\}$$

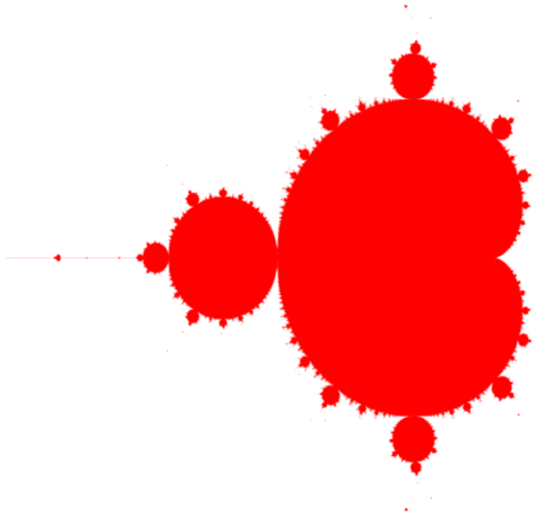
- In fact, it can simulate any Turing machine

  (Turing Complete)

Representing Computation

- Mixes the geometry and computation





Information Measure

- Can define an information measure similar to Kolmogorov Complexity for arbitrary subsets
  - Equivalent to KC for strings.

- Measure by the size of the maps needed in the diagram to represent the subset, in terms of given set of primitive maps

Problems

- Given a diagram and parameters, *render* the subsets to get an approximate dense representation (rendering)
- Given a diagram and a dense data, guess the parameters (parameter finding)
- Given a dense data, find a diagram and parameters that represents the data (pattern discovery)
- Because of the Turing completeness, all these problems are *undecidable* in general

Rendering

- Given the setting →
  and an element, e.g., $x \in X$,
  determine if $x \in s(X)$

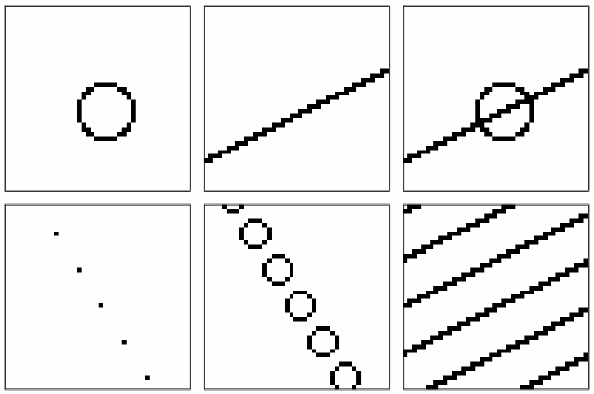
$$\begin{tikzcd} X \arrow[dr, dotted, "\kappa"'] & Y \arrow[l, dotted, "\varphi"'] & Z \arrow[l, dotted, "\psi"'] \\ & W \arrow[u, dotted, "\eta"'] & \end{tikzcd}$$

$s(Z)=\{z\}$

- Recursively search until all conditions are checked

$$x \in s(X) \Leftrightarrow$$

$$\forall \varphi \in \mathrm{in}(X), \begin{cases} \exists y \in s(\mathrm{dm}(\varphi)),\ f(y) = x \ \ [\text{when } \varphi = f] \\ f(x) \in s(\mathrm{dm}(\varphi)) \ \ [\text{when } \varphi = f^{-1}] \end{cases}$$



Rendering (Another approach)

- Create an approximate dense representation for each set. (e.g., pixels)

$$\begin{tikzcd} X \arrow[dr, dotted, "\kappa"'] & & Y \arrow[ll, dotted, "\varphi"'] & & Z \arrow[ll, dotted, "\psi"'] \\ & & W \arrow[u, dotted, "\eta"'] & & \end{tikzcd}$$

$s(Z)=\{z\}$

- Define a binary variable v_x for each node x.
  - $v_x=1$ iff $x \in s(X)$
  - $w_x^\mu=1$ iff $x \in \mu(s(\mathrm{dm}(\mu)))$
- The condition for the $s(X)$ can be converted to logical constraints

$$s(X) = \bigcap_{\mu \in \mathrm{in}(X)} \mu(s(\mathrm{dm}(\mu))) \Leftrightarrow \begin{cases} v_x = \wedge_{\mu \in \mathrm{in}(X)} w_x^\mu \\ w_x^\mu = \vee_{y \in f^{-1}(x)} v_y \quad [\text{when } \mu = f] \\ w_x^\mu = v_{f(x)} \quad [\text{when } \mu = f^{-1}] \end{cases}$$

Rendering (Another approach)

- The constraints are of one of the forms:

$x = y \wedge \ldots \wedge z$	$(x \vee \bar{y} \vee \cdots \vee \bar{z}) \wedge (\bar{x} \vee y) \wedge \cdots \wedge (\bar{x} \vee z)$
$x = y \vee \ldots \vee z$	$(\bar{x} \vee y \vee \cdots \vee z) \wedge (x \vee \bar{y}) \wedge \cdots \wedge (x \vee \bar{z})$
$x = 0$	$\bar{x}$
$x = 1$	x

- They can be converted to the Conjunction Normal Form

$$C_1 \wedge \cdots \wedge C_n$$

$$C_k = x \vee y \vee \bar{z} \vee \cdots \vee w$$

- The problem is thus the Satisfiability (SAT) Problem

MAX SAT

- SAT $\Leftrightarrow$ rendering, parameter finding
  - Rendering: fix the sparse parameter
  - Parameter finding: fix the dense data
- MAX SAT: the optimization version of SAT

$$C_1 \wedge \cdots \wedge C_n, \qquad \text{weight } w_k \text{ for each } C_k$$

$$C_k = \bigvee_{i \in I_k^+} y_i \vee \bigvee_{i \in I_k^-} \bar{y}_i$$

I_k^+ : set of variables appearing unnegated (I_k^- : negated) in clause k

Find the assignment that maximizes $\sum_{k \text{ s.t. } C_k = 1} w_k$

MAX SAT

- Approximation Algorithms

  Yannakakis ’92 (**.75**), Goemans&Williamson ’94 (**.75**), Goemans&Williamson ’95 (**.7584**), Asano&Williamson’02 (**.7846**)

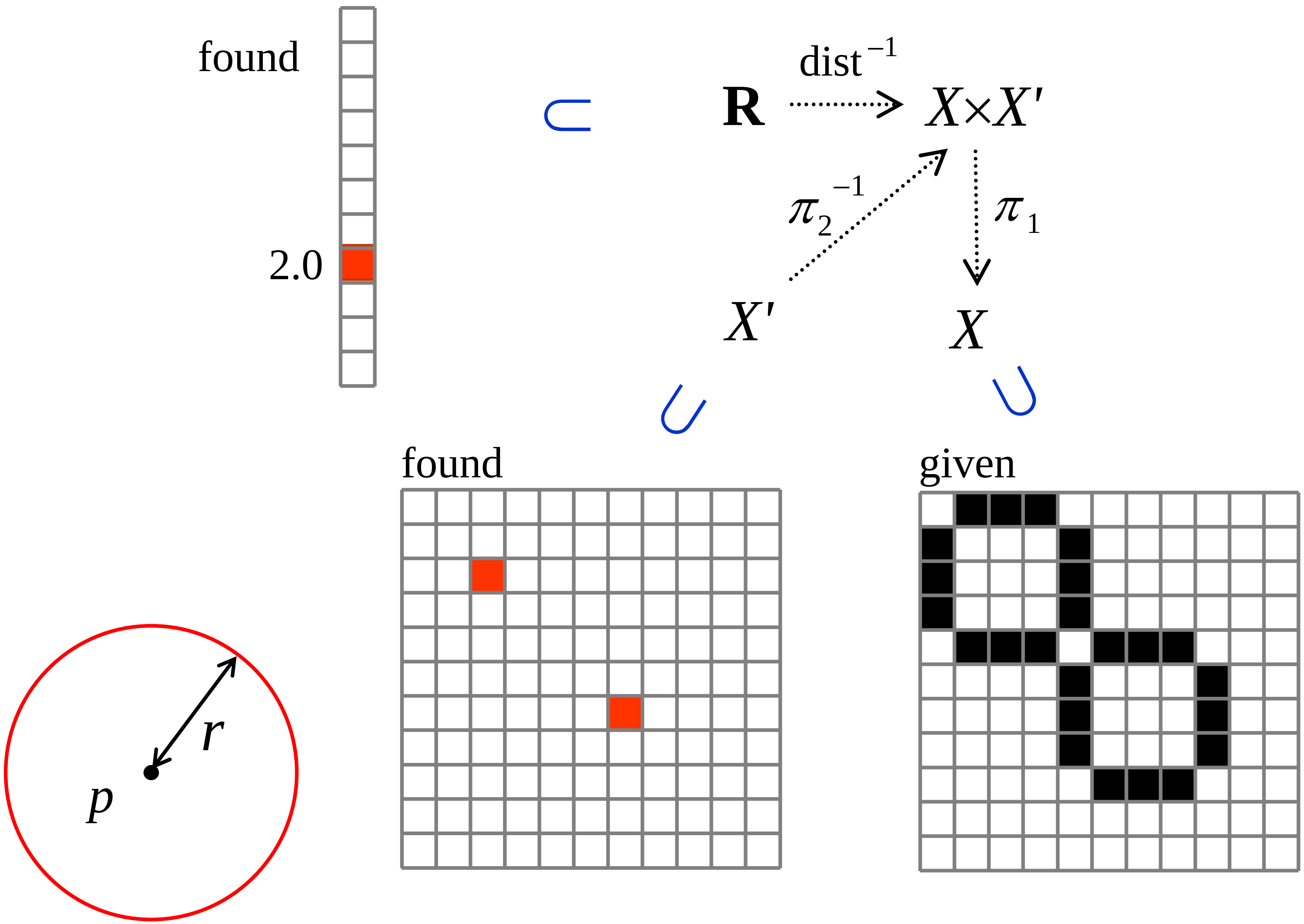
Just to give you a flavor:

relax

$$\text{(IP)} \qquad \begin{aligned} &\text{Max} \quad \sum_{C_k} w_k z_k \\ &\text{subject to}: \sum_{i \in I_k^+} y_i + \sum_{i \in I_k^-} (1 - y_i) \geq z_k \qquad \forall C_k \\ &\qquad y_i \in \{0,1\}, \qquad 0 \leq z_k \leq 1 \end{aligned}$$

Toy Experiment

- Parameter Finding



Conclusion

Introduced a representation of patterns:

- Represents subsets
- Describes structures in terms of the structure as defined by the maps that characterize the space
  - Reflects simplicity relative to the structure → sparse
- Can mix sparse and dense representation
  - Ex. represent lines
    - by a vector and a point → sparse
    - by “pixels” → dense
  - Deals with regular and random parts of the data
- Allows hierarchical and recursive representations
- Problems: rendering, parameter finding, pattern discovery