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

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 $\mathbb{R}^2 \times \text{Color}$
    i.e., image function $I(p)$ on $D \subseteq \mathbb{R}^2 \iff \{(p, I(p)) \mid p \in D\}$

• 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.

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

• Consider an assignment \( s \) to each set of its subset that satisfies

\[
s(S) = \bigcap_{\mu \in \text{in}(S)} \mu (s(\text{dm}(\mu)))
\]

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

\[
\mu : \mathcal{P}(\text{dm}(\mu)) \to \mathcal{P}(\text{cdm}(\mu))
\]

\[
s(X) = \varphi(s(Y)) \\
s(Y) = \psi(s(Z)) \cap \eta(s(W)) \\
s(W) = \varphi(s(X))
\]
Representation

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

Ex.

\[
\begin{align*}
\varphi &: \mathcal{P}(Y) \rightarrow \mathcal{P}(X) \\
\psi &: \mathcal{P}(Z) \rightarrow \mathcal{P}(Y) \\
\eta &: \mathcal{P}(W) \rightarrow \mathcal{P}(Y) \\
\kappa &: \mathcal{P}(X) \rightarrow \mathcal{P}(W) \\
\end{align*}
\]

• Consider an assignment \( s \) to each set of its subset that satisfies

\[
s(S) = \bigcap_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu)))
\]

\[
s(X) = \varphi(s(Y))
\]

\[
s(Y) = \psi(s(Z)) \cap \eta(s(W))
\]

\[
s(W) = \varphi(s(X))
\]

• 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

\[ s(R) = \{ r \} \]
\[ s(X') = \{ p \} \]
\[ s(X) = ? \]

\[ s(S') = \bigcap_{\mu \in \text{int}(S)} \mu(s(dm(\mu))) \]

\[ f: A \rightarrow B \]
\[ f: \mathcal{P}(A) \rightarrow \mathcal{P}(B) \quad A \supset S \mapsto \{ f(x) \mid x \in S \} \subset B \]
\[ f^{-1}: \mathcal{P}(B) \rightarrow \mathcal{P}(A) \quad B \supset S \mapsto \{ x \in A \mid f(x) \in S \} \subset A \]

\( X, X' \): Euclidean plane
\( \text{dist}: X \times X \rightarrow \mathbb{R} \) distance function
\( \pi_1, \pi_2 \): Projection

- Euclidean plane
- Distance function
- Projections
The Simplest Examples

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = \mu s(\text{dm}(\mu)) \]

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]
The Simplest Examples

\[ V \xrightarrow{\pi_1^{-1}} V \times \mathbb{R} \xrightarrow{\text{mult}} V' \]

\[ X' \xrightarrow{\pi_2^{-1}} X \times X' \xrightarrow{\pi_1} X \]

\[ s(V) = \{v\} \]

\[ s(X') = \{p\} \]

\[ s(X) = \? \]

\[ s(V \times \mathbb{R}) = \{(v, c) \mid c \in \mathbb{R}\} \]

\[ s(V') = \{cv \mid c \in \mathbb{R}\} \]

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

\[ s(X) = \{x \mid x = p + cv, c \in \mathbb{R}\} \]

\[ X, X' : \text{Euclidean plane} \]

\[ V, V' : \text{2D vector space} \]

\[ \text{mult}: V \times \mathbb{R} \to V \quad (v, c) \mapsto cv \]

\[ \text{sub}: X \times X \to V \quad (x, y) \mapsto x - y \]

\[ \pi_1, \pi_2 : \text{Projection} \]
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

\[ s(S) = \bigcup_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu))) = \bigcap_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu))) \]

instead of

\[ s(S) = \bigcup_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu))) = \bigcap_{\mu \in \text{in}(S)} \mu(s(\text{dm}(\mu))) \]

\[ s(X) = \varphi(s(Y)) \]
\[ s(Y) = \psi(s(Z)) \cup \eta(s(W)) \]
\[ s(W) = \kappa(s(X)) \]
Formally,

Consider the triple \((\mathcal{I}, \mathcal{I}', \mathcal{M})\) where
\[
\mathcal{I} = \{S_i\}_{i \in I} \text{ is a family of sets indexed by a set } I
\]
\[
\mathcal{I}' \text{ is a subfamily of } \mathcal{I}, \text{ (chosen sets)}
\]
\[
\mathcal{M} = \{\varphi_j\}_{j \in J} \text{ is a family of maps } \varphi_j : \mathcal{P}(S_i) \rightarrow \mathcal{P}(S_k)
\]

Let \(s\) be an assignment to each \(S_i\) in \(\mathcal{I}\) of \(s_i \subset S_i\)
that satisfies
\[
s_i = \bigcup_{j \in J, \varphi_j : \mathcal{P}(S_k) \rightarrow \mathcal{P}(S_i)} \varphi_j(s_k) \quad \text{if } S_i \text{ is in } \mathcal{I}'
\]
and
\[
s_i = \bigcap_{j \in J, \varphi_j : \mathcal{P}(S_k) \rightarrow \mathcal{P}(S_i)} \varphi_j(s_k) \quad \text{if } S_i \text{ is in } \mathcal{I} \setminus \mathcal{I}'
\]
Combination

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]

\[ s(V) = \{ v \} \]

\[ s(X') = \{ q \} \]

\[ R \xrightarrow{\text{dist}^{-1}} X \times X' \]

\[ V \xrightarrow{\pi_1^{-1}} V \times \mathbb{R} \xrightarrow{\text{mult}} V' \]

\[ X' \xrightarrow{\pi_2^{-1}} X \times X' \]

\[ X \subset \subset \text{dist}^{-1} \]

\[ X' \subset \subset \text{sub}^{-1} \]

(p, q, r, v)
Recursive definition

\[ s(V) = \{ v \} \]

\[ s(X') = \{ p \} \]

\[ s(X \times V) = \{(x, v) | x \in s(X)\} \]

\[ s(X) = s(X') \cup \text{add}(s(X \times V)) \]

\[ = \{ p \} \cup \{x + v | x \in s(X)\} \]

\[ \supset \{\cdots, p - 2v, p - v, p, p + v, p + 2v, \cdots\} \]

\[ \pi_1, \pi_2: \text{Projection} \]

\[ X, X': \text{Euclidean plane} \]

\[ V: \text{2D vector space} \]

\[ \text{add}: X \times V \rightarrow X \quad (x, v) \mapsto x + v \]
Recursive definition

\[ V \xrightarrow{\pi_2^{-1}} X \times V \]

\[ X' \xrightarrow{\text{id.}} X \]

\[ s(V) = \{ u, v \} \]

\[ s(X') = \{ p \} \]

\[ s(X) = ? \]

\( X, X' \): Euclidean plane

\( V \): 2D vector space

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

\( \pi_1, \pi_2 \): Projection
Hierarchical definition

\[ V \xrightarrow{\pi_2^{-1}} X \times V \]

\[ s(V) = \{u,v\} \]

\[ X' \xrightarrow{\text{id.}} X \]

\[ s(X') = A \]

\[ s(X) = ? \]

\[ A \subset X \]

\[ X, X' : \text{Euclidean plane} \]

\[ V : 2D \text{ vector space} \]

\[ \text{add}: X \times V \rightarrow X \quad (x, v) \mapsto x + v \]

\[ \pi_1, \pi_2 : \text{Projection} \]
Hierarchical definition

\[ s(\mathbb{R}) = \{ r \} \]

\[ s(X') = \{ p \} \]
Regular and random parts of data

• Example: Same pattern appearing randomly
  – Regularity: the sameness of the pattern
  – Randomness: pattern itself & where the pattern appears
Representing Computation

• Can represent computation

\[
\mathbb{N} \times \mathbb{N} \longrightarrow \mathbb{N} \times \mathbb{N} 
\ni (n, m) \mapsto (n+1, m(n+1))
\]

\[
S(\mathbb{N} \times \mathbb{N}) = \{(0, 1)\} \quad \{(0,1), (1,1), (2,2), (3,6), \ldots, (n,n!), \ldots\}
\]

• 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 \( x \in s(X) \)

• Recursively search until all conditions are checked

\[
x \in s(X) \iff \\
\forall \phi \in \text{in}(X), \begin{cases} 
\exists y \in s(\text{dm}(\phi)), f(y) = x \text{ [when } \phi = f \] \\
 f(x) \in s(\text{dm}(\phi)) \text{ [when } \phi = f^{-1} \]
\]
Rendering (Another approach)

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

• Define a binary variable $v_x$ for each node $x$.
  - $v_x = 1$ iff $x \in s(X)$
  - $w_x^\mu = 1$ iff $x \in \mu(s(dm(\mu)))$

• The condition for the $s(X)$ can be converted to logical constraints

$$s(X) = \bigcap_{\mu \in \text{in}(X)} \mu(s(dm(\mu))) \iff \begin{cases} v_x = \land_{\mu \in \text{in}(X)} w_x^\mu \\ w_x^\mu = \lor_{y \in f^{-1}(x)} v_y \quad \text{[when } \mu = f] \\ w_x^\mu = v_{f(x)} \quad \text{[when } \mu = f^{-1}] \end{cases}$$
Rendering (Another approach)

- The constraints are of one of the forms:
  - $x = y \land \ldots \land z$
  - $x = y \lor \ldots \lor z$
  - $x = 0$
  - $x = 1$

- They can be converted to the Conjuction Normal Form

$$C_1 \land \cdots \land C_n$$

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

- The problem is thus the Satisfiability (SAT) Problem
MAX SAT

• SAT $\iff$ rendering, parameter finding
  - Rendering: fix the sparse parameter
  - Parameter finding: fix the dense data

• MAX SAT: the optimization version of SAT
  \[
  C_1 \land \cdots \land C_n, \quad \text{weight } w_k \text{ for each } C_k
  \]
  \[
  C_k = \bigvee_{i \in I_k^+} y_i \lor \bigvee_{i \in I_k^-} \bar{y}_i
  \]
  $I_k^+$: 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 (0.75), Goemans&Williamson ’94 (0.75),
  Goemans&Williamson ’95 (0.7584), Asano&Williamson’02 (0.7846)

Just to give you a flavor:

relax

\[
\text{Max } \sum_{C_k} \omega_k z_k
\]

(IP) subject to:

\[
\sum_{i \in I^+_k} y_i + \sum_{i \in I^-_k} (1 - y_i) \geq z_k \quad \forall C_k
\]

\[
y_i \in \{0,1\}, \quad 0 \leq z_k \leq 1
\]
Toy Experiment

• Parameter Finding

\[ x \times x' \]
\[ \pi_1 \]
\[ \pi_2 \]
\[ \text{dist}^{-1} \]

\[ R \xrightarrow{\text{dist}^{-1}} X \times X' \]

\[ X' \]
\[ X \]

\[ \bigcup \]

\[ \bigcup \]

\[ p \]
\[ r \]

\[ \text{found} \]
\[ \subset \]
\[ \subset \]
\[ \subset \]

\[ \text{given} \]
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 $\rightarrow$ sparse
- Can mix sparse and dense representation
  - Ex. represent lines
    - by a vector and a point $\rightarrow$ sparse
    - by “pixels” $\rightarrow$ dense
  - Deals with regular and random parts of the data
- Allows hierarchical and recursive representations
- Problems: rendering, parameter finding, pattern discovery