Dependency and Structure in Pattern Recognition

Robert M. Haralick

Computer Science, Graduate Center
City University of New York

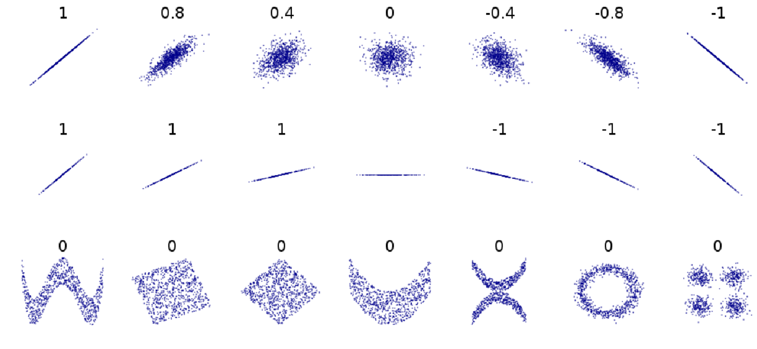
Dependency and Structure

- Pattern Classification
  - Dependency between observations and classes
- Prediction
  - Dependency between the *independent* variables and the response variable
- Compact Representations
  - Dimensionality Reduction
  - Manifold Learning

Characterizing Dependency

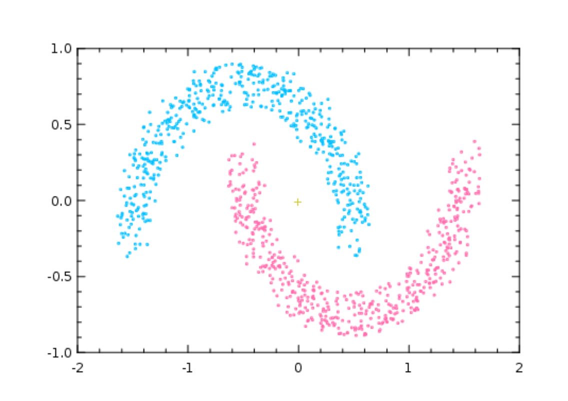
- Measuring Strength of Dependency between/among Variables
- Determining the Dependency Constraint
  - Probabilistic Dependency
  - Coherence: Between Values

Kinds of Dependencies



Graphics from Jason Noble, University of Southampton

Simple Example

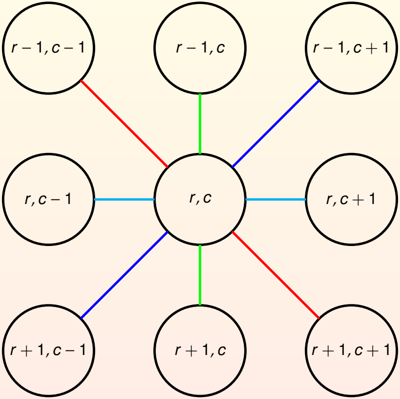


- For some values of X, Y has multiple values
  - Y is not a function of X
- For some values of Y, X has multiple values
  - Y is not a function of X
- There are two non-linear disconnected manifolds

Texture

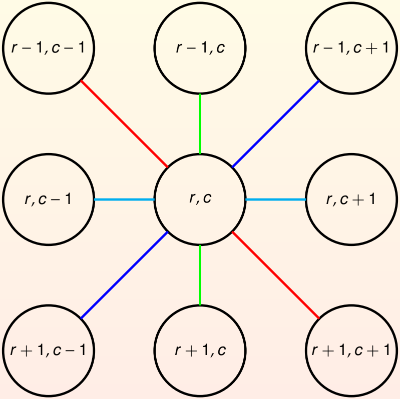
Any patch of an image that shows a texture is a region having a stochastic dependency among the pixel values of the patch.

- Gray level co-occurrence matrix
  - Distance
  - Angle
- Functionals of the co-occurrence matrix can be used as features in distinguishing one texture from another



$$N_1 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r-1,c+1) \text{ or } (u,v) = (r+1,c-1)\}$$
$$N_2 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r,c+1) \text{ or } (u,v) = (r,c-1)\}$$
$$N_3 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r-1,c-1) \text{ or } (u,v) = (r+1,c+1)\}$$
$$N_4 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r-1,c) \text{ or } (u,v) = (r+1,c)\}$$

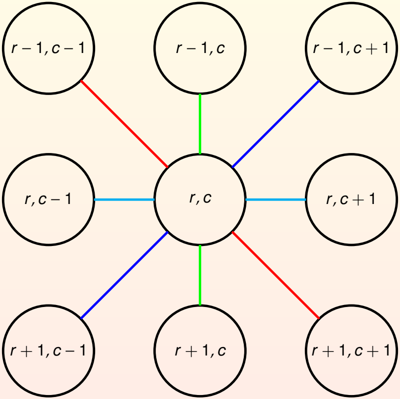
Gray Level Cooccurrence: Major Diagonal



$$N_1 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r-1, c+1) \text{ or } (u,v) = (r+1, c-1)\}$$

$$P_1(i,j) = \frac{\#\{((r,c),(u,v)) \in N_1 \mid \mathcal{I}(r,c) = i \text{ and } \mathcal{I}(u,v) = j\}}{\#N_1}$$

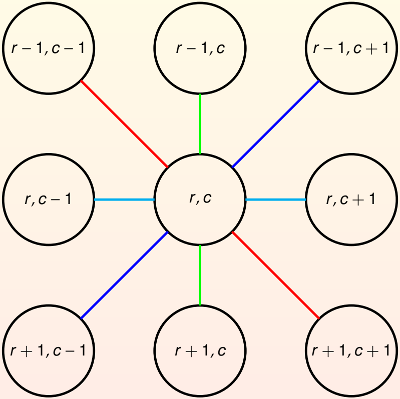
Gray Level Cooccurrence: Left-Right



$$N_2 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r,c+1) \text{ or } (u,v) = (r,c-1)\}$$

$$P_2(i,j) = \frac{\#\{((r,c),(u,v)) \in N_2 \mid \mathcal{I}(r,c) = i \text{ and } \mathcal{I}(u,v) = j\}}{\# N_2}$$

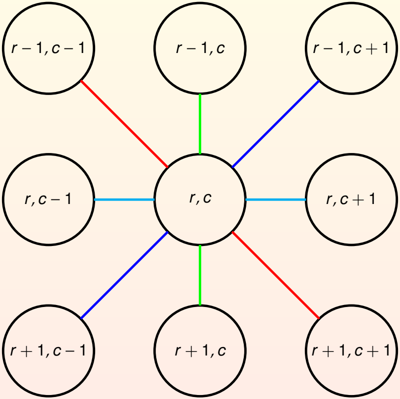
Gray Level Cooccurrence: Minor Diagonal



$$N_3 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r-1,c-1) \text{ or } (u,v) = (r+1,c+1)\}$$

$$P_3(i,j) = \frac{\#\{((r,c),(u,v)) \in N_3 \mid \mathcal{I}(r,c) = i \text{ and } \mathcal{I}(u,v) = j\}}{\#N_3}$$

Gray Level Cooccurrence: Top Bottom

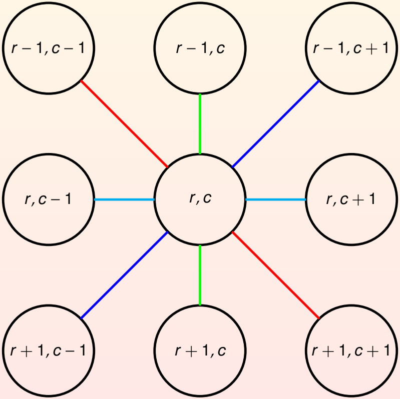


$$N_4 = \{((r,c),(u,v)) \in (R \times C)^2 \mid (u,v) = (r-1,c) \text{ or } (u,v) = (r+1,c)\}$$

$$P_4(i,j) = \frac{\#\{((r,c),(u,v)) \in N_4 \mid \mathcal{I}(r,c) = i \text{ and } \mathcal{I}(u,v) = j\}}{\# N_4}$$

Local Neighborhoods

$$N_1(r,c) = \{(u,v) \in R \times C \mid ((r,c),(u,v)) \in N_1\}$$
$$N_2(r,c) = \{(u,v) \in R \times C \mid ((r,c),(u,v)) \in N_2\}$$
$$N_3(r,c) = \{(u,v) \in R \times C \mid ((r,c),(u,v)) \in N_3\}$$
$$N_4(r,c) = \{(u,v) \in R \times C \mid ((r,c),(u,v)) \in N_4\}$$



Correlation Feature

Because of the symmetry, $P_k(i,j) = P_k(j,i)$ and $I = J$

$$
\begin{aligned}
P_{\{k,row\}}(i) &= \sum_{j=1}^{J} P_k(i,j) \\
P_{\{k,col\}}(j) &= \sum_{i=1}^{I} P_k(i,j) \\
\mu_k &= \sum_{i=1}^{I} i P_{\{k,row\}}(i) = \sum_{j=1}^{J} j P_{\{k,col\}}(j) \\
\sigma_k^2 &= \sum_{i=1}^{I} (i-\mu_k)^2 P_{\{k,row\}}(i) = \sum_{j=1}^{J} (j-\mu_k)^2 P_{\{k,col\}}(j) \\
\rho_k &= \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{(i-\mu_k)(j-\mu_k)}{\sigma_k^2} P_k(i,j) \\
\rho &= \sum_{k=1}^{4} \rho_k \theta_k
\end{aligned}
$$

Entropy Texture Features

$$E_{1k} = \sum_{i=1}^{I}\sum_{j=1}^{J} P_k^2(i,j)$$

$$E_{2k} = -\sum_{i=1}^{K}\sum_{j=1}^{J} P_k(i,j)\log P_k(i,j)$$

$$E_1 = \sum_{k=1}^{K} E_{1k}\theta_k$$

$$E_2 = \sum_{k=1}^{K} E_{2k}\theta_k$$

Robert Haralick, K. Shanmugam, and I. Dinstein,"Textural Features for Image Classification", **IEEE Transactions on Systems, Man, and Cybernetics**, Vol. SMC-3, No. 6,1973, pp. 610-621.

Varieties of Entropy

Let $\alpha > 0$ and $\alpha \neq 1$, then

$$H_\alpha(p_1, \ldots, p_N) = \frac{1}{1-\alpha} \log\left(\sum_{n=1}^{N} p_n^\alpha\right)$$

satisfies the entropy postulates. And

$$\lim_{\alpha \to 1} H_\alpha(p_1, \ldots, p_N) = -\sum_{n=1}^{N} p_n \log p_n$$

Alfrèd Rènyi, "On Measures of Entropy and Information", **Fourth Berkley Symposium on Mathematical Statistics and Probability**, University of California Press, 1961, pp 547-561.

Correlation and Maximal Correlation

$$\rho(X, Y) = E\left[\frac{(X - \mu_x)}{\sigma_x}\frac{(Y - \mu_y)}{\sigma_y}\right]$$

If $E[X] = E[Y] = 0$; and $V[X] = V[Y] = 1$ then

$$\rho(X, Y) = E[XY]$$

Let

$$F = \{f : \mathbb{R} \to \mathbb{R} \mid E[f(X)] = 0; V[f(X)] = 1\}$$
$$G = \{g : \mathbb{R} \to \mathbb{R} \mid E[g(Y)] = 0; V[g(Y)] = 1\}$$

Define Maximal Correlation ρ_{max} by

$$\rho_{max}(X, Y) = \sup_{f \in F, g \in G} E[f(X)g(Y)]$$

H. Gebelein, “Das Statistische Problem der Korrelation als Variations- und Eigenwert-problem und sein Zusammenhang mit der Ausgleichungsrechnung”, **Zeitschrift für Angewwandte Mathematik und Mechanik**, Vol 21, 1941, pp. 364-379.
A. O. Hirschfeld, “A Connection Between Correlation and Contingency”, **Proceedings Cambridge Philosophical Society**, Vol 31, Issue 4, 1935, pp. 520-524.

Maximal Correlation Feature

The normalized joint probability matrix $Q_k = (q_k(i,j))$

$$q_k(i,j) = \frac{P_k(i,j)}{\sqrt{P_{\{k,row\}}(i)}\sqrt{P_{\{k,col\}}(j)}}$$

- The second singular value of Q_k: $\lambda_{k,2}$
- The maximal correlation coefficient: $\rho_{\{max,k\}} = \lambda_{k,2}$

Robert Haralick, K. Shanmugam, and I. Dinstein,"Textural Features for Image Classification", **IEEE Transactions on Systems, Man, and Cybernetics**, Vol. SMC-3, No. 6,1973, pp. 610-621.

Contrast and Inverse Contrast

$$c_k = \sum_{i=1}^{I} \sum_{j=1}^{J} |i-j|^{\beta} P_k(i,j)$$

$$d_k = \sum_{i=1}^{I} \sum_{j=1}^{J} \frac{1}{1+\alpha|i-j|} P_k(i,j)$$

Robert Haralick, K. Shanmugam, and I. Dinstein,"Textural Features for Image Classification", **IEEE Transactions on Systems, Man, and Cybernetics**, Vol. SMC-3, No. 6,1973, pp. 610-621.

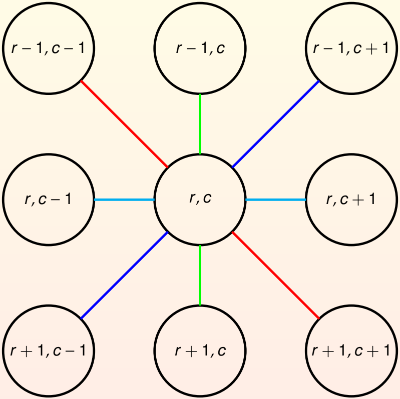
Resolution Preserving Textural Transform Image

- $\mathcal{I}$ Input Image
- P_k, $k = 1, 2, 3, 4$ Cooccurrence Probabilities
- N_k, $k = 1, 2, 3, 4$ Local Neighborhoods
- $\mathcal{J}$ Output Image

$$\mathcal{J}(r,c) = \sum_{k=1}^{4} \sum_{(u,v) \in N_k(r,c)} P_k(\mathcal{I}(r,c), \mathcal{I}(u,v))\theta_k$$

A Resolution Preserving Textural Transform for Images, **Proceedings of the IEEE Computer Society Conference on Computer Graphics, Pattern Recognition, and Data Structure**, San Diego, CA, May 14–16, 1975, pp. 51-61.

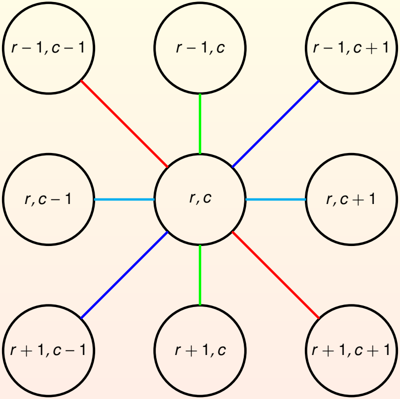
Neighborhood Joint Probability



$$N(r, c) \quad = \quad \{(r, c)\} \cup \bigcup_{k=1}^{4} N_k(r, c)$$

Neighborhood Joint Probability: $P(I(u, v) : (u, v) \in N(r, c))$

Neighborhood Joint Probability



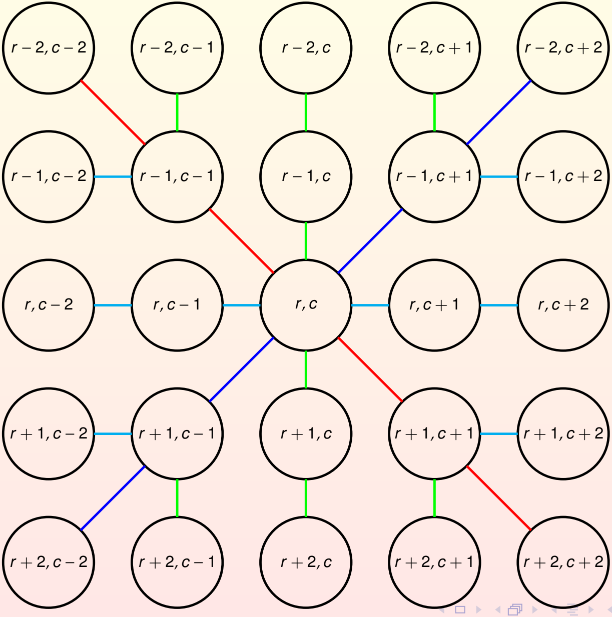
$$P(I(u,v) : (u,v) \in N(r,c)) = P(I(r,c)) \prod_{(u,v)\in N_1(r,c)} P_1(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_2(r,c)} P_2(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_3(r,c)} P_3(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_4(r,c)} P_4(I(u,v) \mid I(r,c))$$

Neighborhood Joint Probability

$$
\begin{aligned}
P(I(r,c), I(u,v) : (u,v) \in N(r,c)) &= P(I(r,c)) \prod_{(u,v)\in N_1(r,c)} P_1(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_2(r,c)} P_2(I(u,v) \mid I(r,c)) \\
&\prod_{(u,v)\in N_3(r,c)} P_3(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_4(r,c)} P_4(I(u,v) \mid I(r,c))
\end{aligned}
$$

$$
\begin{aligned}
J(r,c) = \log(P(I(u,v) : (u,v) \in N(r,c))) &= \log(P(I(r,c))) + \\
&\sum_{(u,v)\in N_1(r,c)} \log(P_1(I(u,v) \mid I(r,c))) + \\
&\sum_{(u,v)\in N_2(r,c)} \log(P_2(I(u,v) \mid I(r,c))) + \\
&\sum_{(u,v)\in N_3(r,c)} \log(P_3(I(u,v) \mid I(r,c))) + \\
&\sum_{(u,v)\in N_4(r,c)} \log(P_4(I(u,v) \mid I(r,c)))
\end{aligned}
$$

Cooccurrence Distance 2 Relations



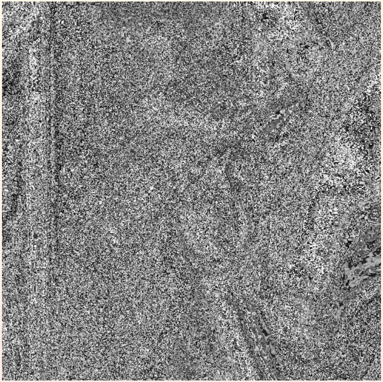
Neighborhood Joint Probability

$$N^2(r,c) = \{(u,v) \mid (u,v) = (r,c) + (i,j), i,j \in \{-2,-1,0,1,2\}\}$$

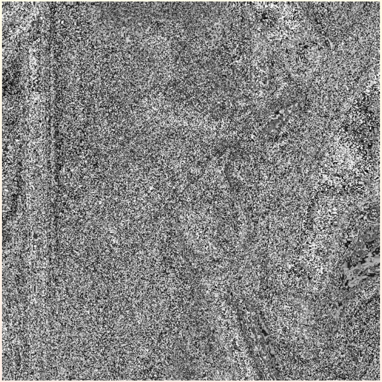
$$P(I(r,c), I(u,v) : (u,v) \in N^2(r,c)) = P(I(r,c)) \prod_{(u,v)\in N_1^2(r,c)} P_1(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_2^2(r,c)} P_2(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_3^2(r,c)} P_3(I(u,v) \mid I(r,c)) \prod_{(u,v)\in N_4^2(r,c)} P_4(I(u,v) \mid I(r,c))$$

$$J(r,c) = \log(P(I(u,v) : (u,v) \in N^2(r,c))) = \log(P(I(r,c))) + \sum_{(u,v)\in N_1^2(r,c)} \log(P_1(I(u,v) \mid I(r,c))) + \sum_{(u,v)\in N_2^2(r,c)} \log(P_2(I(u,v) \mid I(r,c))) + \sum_{(u,v)\in N_3^2(r,c)} \log(P_3(I(u,v) \mid I(r,c))) + \sum_{(u,v)\in N_4^2(r,c)} \log(P_4(I(u,v) \mid I(r,c)))$$

A Texture Image



The Joint Probability Transform



(a) Gray Level Permuted

(b) Joint Probability Transform

Visual Coherence is lost in gray level permuted image.
Probability dependency is the same.
White means the gray level configuration in a 5x5 window in the original image has a high joint probability.

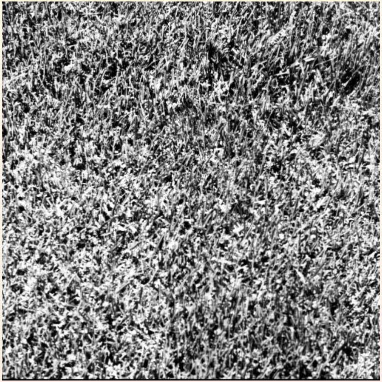
The Joint Probability Transform: Lena

(c) Original

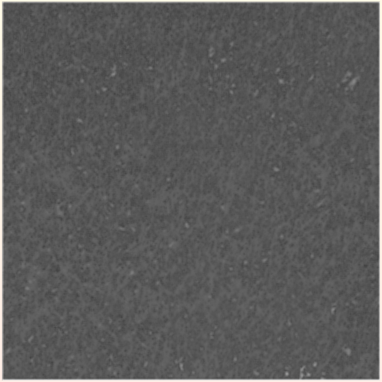
(d) Joint Probability Transform

White means the gray level configuration in a 5x5 window in the original image has a high joint probability.

The Joint Probability Transform: Brodatz



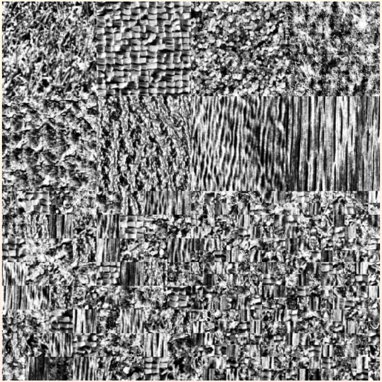
(e) Original



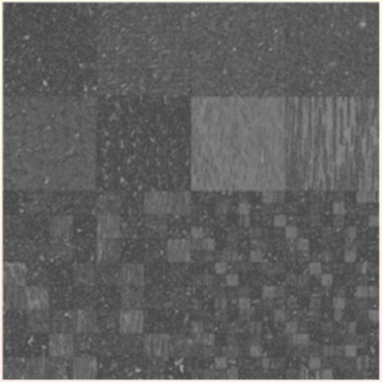
(f) Joint Probability Transform

When the scale of the texture and size of window are comparable, the joint probability is about the same for each window.

The Joint Probability Transform: Brodatz



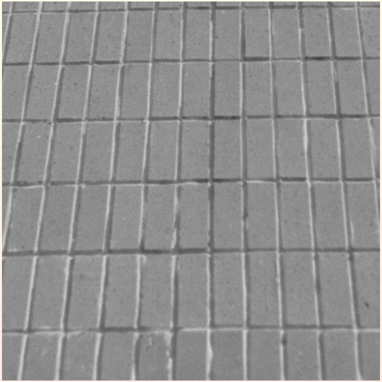
(g) Texture Mosaic



(h) Joint Probability Transform

Within each uniform texture area, the joint probability in a window is nearly the same when the texture scale and the window size are similar.

The Joint Probability Transform: Brodatz

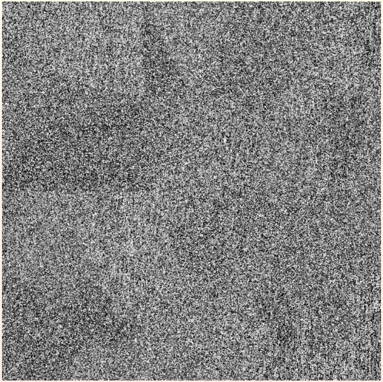


(i) Texture

(j) Joint Probability Transform

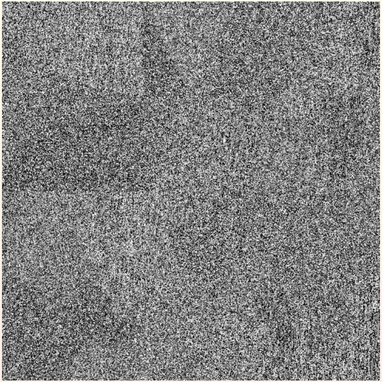
If the texture scale is larger than the windowsize, the joint probability in each window will be low around boundaries.

A Gray Level Permuted Image: Brodatz

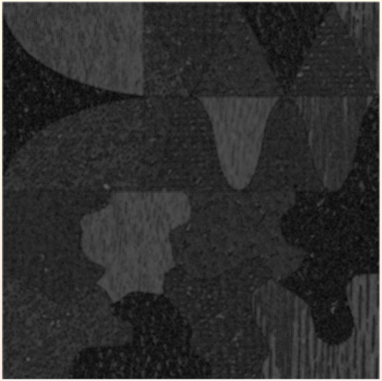


Visual Coherence is lost. Probability Dependency is the same.

The Joint Probability Transform: Brodatz



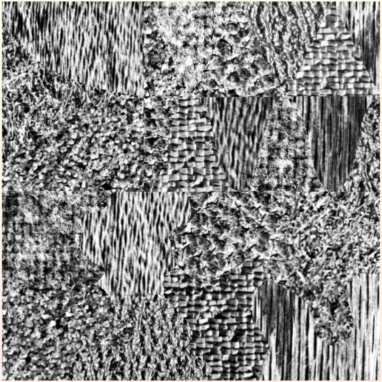
(k) Gray Level Permuted



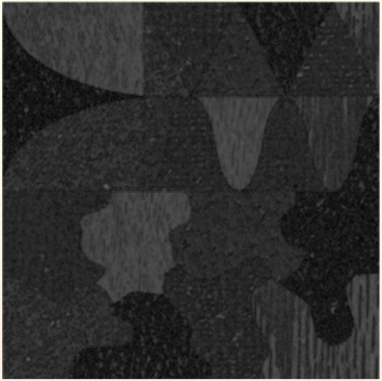
(l) Joint Probability Transform

Within each uniform texture area, the joint probability in a window is nearly the same when the texture scale and the window size are similar.

The Joint Probability Transform



(m) Texture Mosaic



(n) Joint Probability Transform

Within each uniform texture area, the joint probability in a window is nearly the same when the texture scale and the window size are similar.

Conditional Independences
Bound
The Dependencies

Semi-graphoid

Definition

Let I be an index set containing the indexes of all the random variables. Let G be a collection of triples each of whose components are subsets of the index set I. We write $A \perp\!\!\!\perp B \mid C$ if and only if the triple $(A, B, C) \in G$.

G is called a Semi-graphoid if and only if

- Mutual Exclusivity: $(A, B, C) \in G$ implies
  - $A \cap B = \emptyset$, $A \cap C = \emptyset$, $B \cap C = \emptyset$
- Symmetry: $A \perp\!\!\!\perp B \mid C$ if and only if $B \perp\!\!\!\perp A \mid C$
- Decomposition: $A \perp\!\!\!\perp B \cup D \mid C$ implies $A \perp\!\!\!\perp B \mid C$
- Weak Union: $A \perp\!\!\!\perp B \cup C \mid D$ implies $A \perp\!\!\!\perp B \mid C \cup D$
- Contraction: $A \perp\!\!\!\perp B \mid C \cup D$ and $A \perp\!\!\!\perp C \mid D$, imply $A \perp\!\!\!\perp B \cup C \mid D$

A.P. Dawid, “Conditional Independence in Statistical Theory”, **Journal of the Royal Statistical Society**, Vol. 41B, 1979, pp. 1-31

Judea Pearl and Azaria Paz, “Graphoids: A Graph-Based Logic for Reasoning About Relevance Relations”, University of California, Los Angeles, Computer Science Department, CSD-850038, 1985.

Graphoid

Definition

Let I be an index set containing the indexes of all the random variables. Let G be a collection of triples each of whose components are subsets of the index set I. We write $A \perp\!\!\!\perp B \mid C$ if and only if the triple $(A, B, C) \in G$.

G is called a Graphoid if and only if

- Mutual Exclusivity: $(A, B, C) \in G$ implies
  - $A \cap B = \emptyset$, $A \cap C = \emptyset$, $B \cap C = \emptyset$
- Symmetry: $A \perp\!\!\!\perp B \mid C$ if and only if $B \perp\!\!\!\perp A \mid C$
- Decomposition: $A \perp\!\!\!\perp B \cup D \mid C$ implies $A \perp\!\!\!\perp B \mid C$
- Weak Union: $A \perp\!\!\!\perp B \cup C \mid D$ implies $A \perp\!\!\!\perp B \mid C \cup D$
- Contraction: $A \perp\!\!\!\perp B \mid C \cup D$ and $A \perp\!\!\!\perp C \mid D$, imply $A \perp\!\!\!\perp B \cup C \mid D$
- Intersection: $A \perp\!\!\!\perp B \mid C \cup D$ and $A \perp\!\!\!\perp C \mid B \cup D$ imply $A \perp\!\!\!\perp B \cup C \mid D$

Conditional Probabilities and Graphoids

- I is an index set of all the random variables
- $P(X_i : i \in I) > 0$
- G is a collection of triples each of whose components are subsets of the index set I
- $G = \left\{ (A, B, C) \in \mathcal{P}(I)^3 \;\middle|\; \left(\begin{array}{l} A, B \neq \emptyset \\ A, B, C \text{ are disjoint} \\ A \perp\!\!\!\perp B \mid C \end{array} \right. \right\}$

Then G is a graphoid

Conditional Independence Graph: Definition

Definition

A graph (N, E) is called a Conditional Independence Graph of a random variable set $\mathcal{X} = \{X_1, \ldots, X_M\}$ if and only if $N = \{1, \ldots, M\}$, the index set for the variables in $\mathcal{X}$, and

$$E^c = \{\{i, j\} \mid X_i \perp\!\!\!\perp X_j \mid \mathcal{X} - \{X_i, X_j\}\}$$

$\{i, j\}$ not in the edge set means $X_i \perp\!\!\!\perp X_j \mid \mathcal{X} - \{X_i, X_j\}$

Steffen Lauritzen, **Graphical Models**, Clarendon Press, Oxford, 1996.

Separation

Proposition

- $P(x) > 0$
- *$G = (N, E)$ is a conditional independence graph*
- *A, B, C are disjoint subsets of N*
- $A, B, C \neq \emptyset$

If B separates A from C, then $A \perp\!\!\!\perp C \mid B$

Steffen Lauritzen, **Graphical Models**, Clarendon Press, Oxford, 1996.

Conditional Independence: Joint Probability Image

Proposition

- $(u, v), (a, b) \in N^2(r, c)$
- *L is the unique path from (u, v) to (a, b)*
- $(m, n) \in L - \{(u, v), (a, b)\}$
- *$I(u, v)$ is a random variable indexed by (u, v)*
- *$I(a, b)$ is a random variable indexed by (a, b)*
- *$I(m, n)$ is a random variable indexed by (m, n)*

Then

$$I(u, v) \perp\!\!\!\perp I(a, b) \mid I(m, n)$$

Triangulated Graphs

Theorem

If a graph G is triangulated graph and $C_1, \ldots, C_K$ are the cliques of G put in running intersection order with separators $S_2, \ldots, S_K$,

$$S_k = C_k \bigcap \left(\bigcup_{i=1}^{k-1} C_i \right), k = 2, \ldots, K$$

then

$$P(x_1, \ldots, x_N) = \frac{\prod_{k=1}^{K} P(x_i : i \in C_k)}{\prod_{k=2}^{K} P(x_i : i \in S_k)}$$

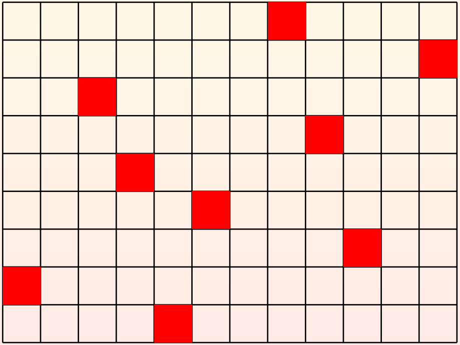
Steffen Lauritzen, **Graphical Models**, Clarendon Press, Oxford, 1996.

N-Tuple Method

- Developed For Printed Character Recognition
- Each character is contained in an image of $M \times N$ pixels
- Each pixel is a binary 1 or a binary 0
- Designed for table lookup hardware

W.W. Bledsoe and I. Browning, “Pattern Recognition and Reading by Machine”, **Proceeding Eastern Joint Computer Conference**, Boston, 1959, pp. 232-255.

N-Tuple Method



N-Tuple Method

- Small number of pixel positions are randomly selected
- Each of these positions has a binary 0 or a binary 1
- Concatenate all the binary values to form a binary number
- Use this number to form an address in memory
- Have as many memory arrays as character classes
- Have multiple sets of such randomly selected pixel positions

N-Tuple Method

- M pattern sets of randomly selected pixel positions
- K character classes
- T_{mk} lookup table for pattern set m and class k
- $T_{mk}(b_m)$ holds a binary 1 if a character in the training set of class k has the binary number b_m for the m^{th} pattern set
- A printed character produces M binary numbers $b_1, \ldots, b_M$
- Compute
  - $f_k = \min_{m=1}^{M} T_{mk}(b_m)$
  - $f_k = \sum_{m=1}^{M} T_{mk}(b_m)$
- Assign the character to unique class c_k, if there is one, for which $f_k > 0$ is highest
- Otherwise reserve decision

Relations

- Each of the possible pixel positions is a variable
- Let $X_1, \ldots, X_N$ be the N variables
- Let L_n be the possible values variable X_n can take
- Let R be the training set for one class

$$R \subseteq \bigtimes_{n=1}^{N} L_n$$

N-ary Relation

Definition

If I is an index set and $R \subseteq \bigtimes_{i \in I} L_i$, then we say (I, R) is an Indexed N-ary Relation on the range sets indexed by I.

Relation Join

Definition

Let I, J, K be index sets with $K = I \cup J$. Let $R \subset \bigtimes_{i \in I} L_i$ and $S \subset \bigtimes_{j \in J} L_j$ Then the Relation Join of (I, R) with (J, S) is denoted by $(I, R) \otimes (J, S) = (K, T)$ where

$$T = \{t \in \bigtimes_{k \in K} L_k \quad | \quad \pi_I(K, t) \in (I, R) \text{ and } \pi_J(K, t) \in (J, S)\}$$

N-Tuple Method

The set of measurement tuples that could be assigned to a class c, is the relation join of the tables associated with class c.

Theorem

$$\{([N], x) \mid \pi_{J_m}([N], x) \in (J_m, T_{mc}), m = 1, \ldots, M\} = \otimes_{m=1}^{M} (J_m, T_{mc})$$

- (J_m, T_{mc}) is an indexed relation
- Defined on index set J_m associated with pattern set m
- T_{mc} Contains all binary tuples from pattern set m that were class c

N-Tuple Method

- M pattern sets of randomly selected pixel positions
- K character classes
- T_{mk} lookup table for pattern set m and class k
- $T_{mk}(b_m)$ holds the fraction of times a character in the training set of class k has the binary number b_m for the m^{th} pattern set
- A printed character produces M binary numbers $b_1, \ldots, b_M$
- Compute
  1. $f_k = \min_{m=1}^{M} T_{mk}(b_m)$
  2. $f_k = \sum_{m=1}^{M} log T_{mk}(b_m)$
- For (1) Assign the character to unique class c_k, if there is one, for which $f_k > 0$
- For (2) Assign the character to unique class c_k, if there is one, for which $f_k > -\infty$
- Otherwise reserve decision

N-Tuple Method Sum of Logs

$$f_k = \sum_{m=1}^{M} log T_{mk}(b_m)$$

This is equivalent to

$$\log P_c(x_1, \ldots, x_N) = \sum_{k=1}^{K} \log P_c(x_i : i \in C_k)$$

The class conditional independence assumption assumed here is surely wrong.

N-Tuple Method Using Graphical Model

$$P_c(x_1, \ldots, x_N) = \frac{\prod_{k=1}^{K} P_c(x_i : i \in C_k)}{\prod_{k=2}^{K} P_c(x_i : i \in S_k)}$$

$$= P_c(x_i : i \in C_1) \prod_{k=2}^{K} P_c(x_i : i \in C_k - S_k \mid x_j : j \in S_k)$$

$$\log P_c(x_1, \ldots, x_N) = \log P_c(x_i : i \in C_1) + \sum_{k=2}^{K} \log P_c(x_i : i \in C_k - S_k \mid x_j : j \in S_k)$$

Conditional No Influences
Bound
The Dependencies

Conditional No Influence

- $L_1 = \{a_1, a_2, a_3\}$
- $L_2 = \{c_1, c_2\}$
- $L_3 = \{b_1, b_2, b_3\}$
- $I = \{1, 2, 3\}$

$R \subset \times_{i \in I} L_i$		
1	**2**	**3**
a_1	c_1	b_1
a_1	c_1	b_2
a_1	c_1	b_3
a_2	c_1	b_1
a_2	c_1	b_2
a_2	c_1	b_3
a_1	c_2	b_4
a_1	c_2	b_5
a_3	c_2	b_4
a_3	c_2	b_5

1 has no influence on **3** given **2**

Conditional No Influence

- $L_1 = \{a_1, a_2, a_3\}$
- $L_2 = \{c_1, c_2\}$
- $L_3 = \{b_1, b_2, b_3\}$
- $I = \{1, 2, 3\}$
- $J = \{1, 2\}$
- $K = \{2, 3\}$
- $(I, R) = \pi_J(I, R) \otimes \pi_K(I, R)$
- $J - K \perp\!\!\!\perp K - J \mid J \cap K$

Theorem

Let $G = \{(A, B, C) \in \mathcal{P}^3(I) \mid A \perp\!\!\!\perp B \mid C\}$. Then G is a Semi-graphoid.

Robert Haralick, Ligon Liu, Evan Misshula, "Relation Decomposition: The Theory", **International Conference on Machine Learning and Data Mining**, (MLDM), 2013, New York, 2013 pp. 311-324.

Relation Join Factor Properties

If a given relation is factored into a decomposition of N factor relations, then grouping these factor relations into two possibly overlapping groups will also factor the given relation.

Proposition

Let $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$ *and* $S \cup T = \{1, \ldots, N\}$. $S, T \neq \emptyset$, *Then*

$$\pi_{\cup_{s \in S} M_s}(M, R) \otimes \pi_{\cup_{t \in T} M_t}(M, R) = (M, R)$$

Robert Haralick, Ligon Liu, Evan Misshula, "Relation Decomposition: The Theory", **International Conference on Machine Learning and Data Mining**, (MLDM), 2013, New York, 2013 pp. 311-324.

Relation Join Factor Properties

Corollary

Let $(M, R) = \otimes_{n=1}^{N} \pi_{M_n}(M, R)$. *Define* $\mathcal{T} = \{T \mid \text{ for some } S \subset [N], S \neq \emptyset, T = \cup_{s \in S} M_s\}$, *then* $U, V \in \mathcal{T}$ *implies*

$$U - V \perp\!\!\!\perp V - U \mid U \cap V$$

Robert Haralick, Ligon Liu, Evan Misshula, "Relation Decomposition: The Theory", **International Conference on Machine Learning and Data Mining**, (MLDM), 2013, New York, 2013 pp. 311-324.

Cliques and No Influence Pair of Sets

$\mathbf{M} = \{1,2,3,4,5,6,7,8,9\}$

$(\mathbf{M},\mathbf{R}) = \otimes_{n=1}^{8} \pi_{\mathbf{M_n}}(\mathbf{M},\mathbf{R})$

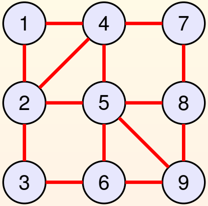
$\mathbf{I} = \{1,2\}$, $\mathbf{J} = \{8,9\}$, $\mathbf{I} \cap \mathbf{J} = \emptyset$

$\mathbf{M} - (\mathbf{I} \cup \mathbf{J}) = \{3,4,5,6,7\}$

$\mathbf{I} \perp\!\!\!\perp \mathbf{J} \mid \mathbf{M} - (\mathbf{I} \cup \mathbf{J})$

$\mathbf{I} \cup (\mathbf{M} - (\mathbf{I} \cup \mathbf{J})) = \mathbf{M} - \mathbf{J}$

$\mathbf{J} \cup (\mathbf{M} - (\mathbf{I} \cup \mathbf{J})) = \mathbf{M} - \mathbf{I}$



$(\mathbf{M},\mathbf{R}) = \pi_{\mathbf{M-I}}(\mathbf{M},\mathbf{R}) \otimes \pi_{\mathbf{M-J}}(\mathbf{M},\mathbf{R})$

$\mathbf{I} \times \mathbf{J} \subseteq \mathbf{M} \times \mathbf{M} - \cup_{n=1}^{8} \mathbf{M_n} \times \mathbf{M_n}$

$\mathbf{S} = \{\mathbf{n} \mid \mathbf{M_n} \subseteq \mathbf{M} - \mathbf{J}\} = \{1,2,5,6,7\}$

$\mathbf{T} = \{\mathbf{n} \mid \mathbf{M_n} \subseteq \mathbf{M} - \mathbf{I}\} = \{3,4,6,7,8\}$

$\mathbf{M} - \mathbf{J} = \mathbf{M_1} \cup \mathbf{M_2} \cup \mathbf{M_5} \cup \mathbf{M_6} \cup \mathbf{M_7} = \cup_{\mathbf{s} \in \mathbf{S}} \mathbf{M_s}$

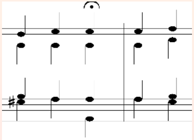
$\mathbf{M} - \mathbf{I} = \mathbf{M_3} \cup \mathbf{M_4} \cup \mathbf{M_6} \cup \mathbf{M_7} \cup \mathbf{M_8} = \cup_{\mathbf{t} \in \mathbf{T}} \mathbf{M_t}$

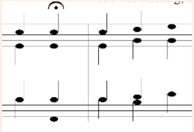
(o) Influence Graph

Clique Symbol	Cliques	$\mathbf{M} - \mathbf{J}$ $\{1,2,3,4,5,6,7\}$	$\mathbf{M} - \mathbf{I}$ $\{3,4,5,6,7,8,9\}$
M_1	$\{1,2,4\}$	1	0
M_2	$\{2,4,5\}$	1	0
M_3	$\{5,6,9\}$	0	1
M_4	$\{5,8,9\}$	0	1
M_5	$\{2,3\}$	1	0
M_6	$\{3,6\}$	1	1
M_7	$\{4,7\}$	1	1
M_8	$\{7,8\}$	0	1

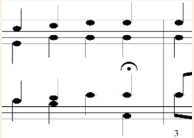
(p) Cliques

Bach Choral BWV26


(q) Tuple 2


(r) Tuple 3


(s) Tuple 4


(t) Tuple 5

Xiuyan Ni, Ligon Liu, Robert Haralick, ”Music Generation With Relation Join”, **12th International Symposium on Computer Music Multidisciplinary Research**, São Paulo, 2016, pp. 286-296.

No Influence

- R Set of 5-tuples obtained from Music Corpus
- $I = \{1, 2, 3, 4, 5\}$
- $J = \{3, 4, 5, 6, 7\}$
- Construct the join $(K, S) = (I, R) \otimes (J, R)$

Proposition

If $(K, S) = (I, R) \otimes (J, R)$, $I - J \neq \emptyset$, *and* $J - I \neq \emptyset$
Then,

$$I - J \perp\!\!\!\perp J - I \mid I \cap J$$

$$\{1, 2\} \perp\!\!\!\perp \{6, 7\} \mid \{3, 4, 5\}$$

Joining The Tuples

$$
\begin{aligned}
I_1 &= \{1, 2, 3, 4, 5\} \\
I_2 &= \{3, 4, 5, 6, 7\} \\
&\vdots \\
I_n &= \{2n-1, 2n, 2n+1, 2n+2, 2n+3\} \\
&\vdots
\end{aligned}
$$

- Construct $(K, S) = \otimes_{n=1}^{N} (I_n, R)$
- Sample tuples from (K, S) to listen to

Bach Generation By Relation Join

- 202 Bach Chorals Midi Files Taken from Music21
- Sequence of 5 Chords and Durations Constitute A Tuple
- Relation Join Requires Overlap of 3 Chords
- First Chord of Join Must be the I Chord of the Key
- Last Chord of the Join Must be a long duration I Chord of the Key

A Bach Choral

Random Sample Generated Bach Choral

- 4 Chords and Overlap 2 generate over 24 million sequences

Xiuyan Ni, Ligon Liu, Robert Haralick, "Music Generation With Relation Join", **12th International Symposium on Computer Music Multidisciplinary Research**, São Paulo, 2016, pp. 286-296.