Consider a set of n nodes: $i = 1, ..., n$.

Let $\{\Lambda_j\}$ be a set of m labels: $\lambda_1, ..., \lambda_m$ associated with each node i.

Define the Constraint Set as follows:
$\Lambda_{ij} = \{(\lambda_i, \lambda_j^{'})$: label λ at node i is "compatible" with label $\lambda^{'}$ at node j $\}$.

The assignments are given by:
$p_i(\lambda) = 1$ if λ is associated with node i.
$p_i(\lambda) = 0$ otherwise.

The compatibility constraints are given by:
$R_{ij}(\lambda, \lambda^{'}) = 1$ if $(\lambda, \lambda^{'}) \in \Lambda_{ij}$.
$R_{ij}(\lambda, \lambda^{'}) = 0$ otherwise.

"Support" for label λ at node i:

$$s_i(\lambda) = \sum_{j(\text{neighbors})} \max_{\lambda' \in \Lambda_j} \{R_{ij}(\lambda, \lambda') p_j(\lambda')\}$$

Observe $s_i(\lambda)$ is just the number of neighbors with at least one compatible label (the assignments and the constraints are binary valued).

Label Discarding Rule (applied iteratively):
Discard a label λ at node i if there exists a neighbor j such that $(\lambda, \lambda') \notin \Lambda_{ij}$ for all λ' currently assigned to node j.

Label λ at node i is retained only when each neighbor j has at least one compatible label λ'.

Applying the label discarding rule, any surviving label must λ at i must have support:

$$
\begin{aligned}
s_i(\lambda) &= \text{number of neighbors of } i \\
s_i(\lambda) &\geq s_i(\lambda')
\end{aligned}
$$

A *consistent* labelling is obtained when no further labels can be discarded.

Let the compatibility constraints and the assignments be real-valued.

Assignments:

$$0 \leq p_i(\lambda) \leq 1 \text{ for all } i, \lambda$$

$$\sum_{\lambda=1}^{m} p_i(\lambda) = 1 \text{ for all } i = 1, ..., n$$

Compatibility Constraints:

$$\begin{aligned} r_{ij}(\lambda, \lambda^{'}) &> 0 \text{ for compatible labels} \\ &= 0 \text{ for noninteracting labels} \\ &< 0 \text{ for incompatible labels} \end{aligned}$$

Consistency?

Space of Weighted Labeling Assignments: $\mathcal{K}$

Let the variables $p_i(1), ..., p_i(m)$ compose an m-vector $\widehat{p_i}$. The concatenation of these vectors, $\widehat{p_1}$;$\widehat{p_2}$;...;$\widehat{p_n}$ form an assignment vector $\widehat{P} \in \mathcal{R}^{nm}$

The space of *weighted* labeling assignments $\mathcal{K}$, is given by:

$$\begin{aligned}
\mathcal{K} = \{\widehat{P} \in \mathcal{R}^{nm} \quad : \quad & \widehat{P} = (\widehat{p}_1, ..., \widehat{p}_n); \\
& \widehat{p}_i = (p_i(1), ..., p_i(m)) \in \mathcal{R}^m; \\
& 0 \leq p_i(\lambda) \leq 1, \text{ for all } i, \lambda \\
& \sum_{\lambda=1}^{m} p_i(\lambda) = 1, \text{ for all } i \\
\}
\end{aligned}$$

The space of *unambiguous* labeling assignments $\mathcal{K}^*$, is given by the above, with the restriction that every $p_i(\lambda)$ is either 0 or 1, implying that each node has exactly one label.

"Support" for label λ at node i by the assignment $\hat{P}$ is given by:

$$s_i(\lambda; \hat{P}) = \sum_{j=1}^{n} \sum_{\lambda'=1}^{m} r_{ij}(\lambda, \lambda') p_j(\lambda')$$

where the coefficients $\{r_{ij}(\lambda, \lambda')\}$ comprise a matrix of compatibilities.

An unambiguous labeling assignment is consistent if the support for the label assigned to every node is greater than or equal to the support for any other label at that node (with the assignment fixed).

$\hat{P}$ is consistent in $\mathcal{K}^*$ provided that

$$\begin{aligned} s_1(\lambda_1; \hat{P}) &\geq s_1(\lambda; \hat{P}) \quad 1 \leq \lambda \leq m \\ &\ldots \\ s_n(\lambda_n; \hat{P}) &\geq s_n(\lambda; \hat{P}) \quad 1 \leq \lambda \leq m \end{aligned}$$

A restatement of this condition for consistency:

$$\sum_{\lambda=1}^{m} p_i(\lambda) s_i(\lambda; \hat{P}) \geq \sum_{\lambda=1}^{m} v_i(\lambda) s_i(\lambda; \hat{P}) \quad i = 1, ..., n$$

for all *unambiguous* labelings $\hat{V} \in \mathcal{K}^*$.

This leads to an analogous condition for *weighted* labeling assignments.
$\hat{P}$ is consistent in $\mathcal{K}$ provided that

$$\sum_{\lambda=1}^{m} p_i(\lambda) s_i(\lambda; \hat{P}) \geq \sum_{\lambda=1}^{m} v_i(\lambda) s_i(\lambda; \hat{P}) \quad i = 1, ..., n$$

for all *weighted* labelings $\hat{V} \in \mathcal{K}$.

The goal of the relaxation is to find a consistent labeling, which is equivalent to solving a variational inequality.

Theorem: A labeling $\hat{P} \in \mathcal{K}$ is consistent if and only if

$$\sum_{i,\lambda,j,\lambda'} r_{ij}(\lambda, \lambda')p_j(\lambda')[v_i(\lambda) - p_i(\lambda)] \leq 0$$

for all $\hat{V} \in \mathcal{K}$.

(The proof is straightforward).

Can the inequality be solved using optimization?

Define the average local consistency:

$$A(\hat{P}) = \sum_{i=1}^{n} \sum_{\lambda} p_i(\lambda) s_i(\lambda)$$

$$= \sum_{i,\lambda} \sum_{j,\lambda'} r_{ij}(\lambda, \lambda^{'}) p_j(\lambda^{'}) p_i(\lambda)$$

Observe that maximizing $A(\hat{P})$ is not the same as finding a consistent labeling, such that

$$\sum_{\lambda} p_i(\lambda) s_i(\lambda)$$

is maximal for all $i = 1, ..., n$.

Why?

Because the n quantities are NOT independent.

Theorem: Let the matrix of compatibilities be symmetric, i.e., $r_{ij}(\lambda, \lambda^{'}) = r_{ji}(\lambda^{'}, \lambda)$ for all $i, j, \lambda, \lambda^{'}$. If $A(\hat{P})$ attains a local (relative) maximum at $\hat{P} \in \mathcal{K}$, then $\hat{P}$ is a consistent labeling.

Hence, for symmetric compatibilities the problem of finding consistent labelings reduces to that of finding local maxima of an objective function $A(\hat{P})$.

For non-symmetric compatibilities local maximizers of $A(\hat{P})$ will NOT correspond to consistent labelings in general. In such cases, a variational method can be used to solve the system of inequalities.